Spring framework overview

00:00:00
welcome back aliens my name is Av ready and in this video we'll talk about spring framework and microservices and in this particular video we'll learn about spring framework how spring framework Works what is spring MVC basically how do you build a web application using spring how do you work with database connectivity how do you create a project using spring framework and how to connect with database as well not just SQL databases but no SQL which is mongod DB and then we'll move towards

00:00:26
Spring aop also how do you secure your application where important right and that's why we'll talk about sing security and then we'll move towards microservices now of course when you move from monic application to microservices how do you create a project in monic how do you break it down into microservices we are going to learn everything in this particular video now when you look at this video the spring part we have used spring five there so spring five and spring boot 2.5 up version and for micro Services we

00:00:57
have used version three so basically just follow the video and go with a specific version which I which I'm using there most of the content will relevant for spring 6 as well it's just that in Spring Security there have been a lot of changes recent time uh so at that point just just try to check for the recent code and I will try to push the recent code of Spring Security 6 in the description so make sure that you check the description for the code as well and we'll try to push the recent one as well

00:01:24
with the spring six and the older one which is spring five and that's specific for the Spring Security other will work as it is now I hope you will enjoy this course of course a lengthy video but once you complete this entire stuff you will know how to build application using spring and microservices and also the prerequisite for this course is Java of course if you know Java you can start with this course otherwise look for the co Java video on this channel of course Co Java jdbc something uh which is

00:01:52
prerequisite for this course and if you know the concept of Serv JSP it will be helpful if not still you can go for this course and somewhere you have to it will take more time for you for for you to connect what I'm trying to say if you don't understand what the savet is but I would recommend if you don't know savet check out the video in description you will find a code Java and Sal videos there so I hope you'll enjoy [Music] this now when you say spring it basically means spring framework right

00:02:19
we're talking about a framework here and spring is one of the best framework available for Java yes we had different Frameworks before example for web development we were using STS in the Enterprise Market we would happy with ejb and then on the other hand we have database side where we were happy with jpa but then now with the help of spring you can achieve everything the

one thing which is amazing about spring is it focus on pojos now remember when you talk about ajb it was all about entities

00:02:45
right they were good you know you can build huge applications which are stable scalable using ejb but then still the components which we were using is very heavy in terms of the size or the services it uses but what if I say you can do everything with the help of poos now if you don't know what this pojo means you know if you don't know what services means that's completely fine just for this course you will get everything but then if you know about poos yes we can use poos in spring now

00:03:13
spring is also the combination of multiple projects it's not just one project which you'll be using example let's say if you want to achieve a very common feature of spring which is dependency injection we have a separate module for it if you want to work on web if you want to make a web application we have a separate module for it and of course one of our favorite if you want to build a rest application we have a seate module what if I say spring also can be used for security that's awesome

00:03:38
right as a developer we are focusing more on application development we are not focusing on the security part of it and spring will help you there as well if you want to connect with database with the help of some jpa spring has a different module so spring is basically a collection of module it has right and then we'll be going through most of this modu so we'll be talking about spring code which will help you for a dependency injection we'll talk about spring MVC which will help you for web

00:04:01
development we'll talk about spring rest to make rest apis and this goes [Music] on so before going for the actual modules let's talk about one important thing which is the official website of spring now if you go to official website let's go to Google and search for spring and the moment you search for spring it will give you this official website which is spring.io just click on that so you can see says Spring by pyot so pyot is basically company who has designed spring and you can use it for free uh

00:04:34
that's great that's an open source project which you have now if you click on Spring projects you can see we have lot of projects here which we have talked about before we have spring boot again that's a surprise for you what is spring boot will not reveal that as of now we have spring framework if you want to work with data flow we have that as well if you want to work on Spring Cloud we have spring Cloud we have spring data and then this is very important Spring Security and then you can also build

00:04:59
spring web services right which is your soap and least goes on there are so many features which we can use here of course we we cannot be covering everything in this course we'll be focusing on most important part which is spring framework so of course we have our focus is this one with this we'll also cover spring Boot and spring data will be also there and then one of my favorite which is Spring Security okay that's the official website and the amazing thing about spring is you know not just spring it's

00:05:24

about Java we have amazing documentation same goes for Java docs we have in fact is show of referring any book you can refer Java docs and you can solve your queries we have huge community in the same way we have spring documents as well so if you can just go to the guides do we have here uh so you can see you will get a lot of help from this otherwise you can just click on this and search for spring docs and you can see it gives you spring documentation as well you can click on guides and this is

00:05:52
where you will get the reference documentation just click on this in fact they will also give you a performance so let's say if you want to read your book in mobile so you can also use UB uh in fact if you want to get that you can simply search for spring framework ePub and this will get to the official website you can just go here okay they're giving you 4.3 5. is also also available if you click on the first one you can see that we have the reference document dat which is ePub of spring five that's amazing

00:06:22
right in fact one of I'm a big fan of this documentation at least for spring you can refer everything on book uh the ebook which I'm talking about otherwise you can just refer online you can see we have online documentation let's say if you want to know about spring Cloud just click on Spring cloud and you can see it gives you all the options you can click on documentation and you will get it so you can see I'm clicking on the documentation you can see we got single HTML open that and you can read the

00:06:46
entire documentation so that's how you can refer documentation yes there are some books available as well on Amazon on different website of course so you can just go there and you can buy any book I will also try to list my favorite books available for spring but then as of now the spring documentations is awesome and with this videos of course now what are the pr physics for spraying the first thing you should know is Java of course right so if you don't know Java how can you work on Spring

00:07:12
framework in fact any framework on Java so the Coe Java part is very important uh you should know what is oops you should know what is exception handling what is threads and all those basic concept of course Loops as well the next thing you should know is database connectivity So when you say database it can be Mas with Java it should be maybe post with Java so whatever database you have worked on with Java that jdbc is very important so if you know that that will be amazing the next two is very

00:07:37
important the first one is salet and jsb now of course when you work on Spring MVC it's of MVC model now you'll be thinking okay we have done with Serv JSP why you have to learn old Technologies what if I say in Java framework you have we normally work on Tomcat right so Tomcat is Serv container which simply means even if you are running spring framework on Tomcat ultimately everything will be converted into servlets so having their knowledge of servet and JSP will be awesome so not everything but Basics are very important

00:08:09
example you should know how to send data from client to server then what is HTTP request object HTTP response object all those basic concept that's it the next thing you should be knowing is omm framework now it can be hybernate ibes or simple GPA everything will be fine so I would suggest if

you have not seen hybernate videos it is available on YouTube for free just go there and you can watch hyate videos so that's the basic thing you'll be required here that's it nothing else uh yes there are

00:08:34
some optional concept if you know that will be amazing you know how internet works the protocol works then the next thing is you know we have for sping security we should be knowing about ldap o the basics of it of course that will be advantaged but again once we start with a particular module in every module I'll be talking about some basics of it what are things required to for you to learn so you can just pause the videos and can go back and Google it right it's not that difficult even if you don't

00:08:58
know KJ that's fine you can just just pause this video and know pause the series go back to my COA videos watch some of them which is really available on this website just go there and you can learn basic Co Jaa and you can come back here that's completely fine so that's the prerequisite you should be having and that's it that's it [Music] everyone now it's time to talk about the requirement in terms of software of course right if you want to write your spring code where will write it in fact

00:09:24
the we have the simple way right you can open notepad type your code and you can run it in command line we can do that but hold on we will be building big projects right so of course if you're learning spring framework in future you will be building amazing projects and that's where if you have a good ide with you it will be an awesome thing right and that's where we have lot of Ides available you can use net beans which is everyone's favorite you can use Eclipse which is industrywide famous you can

00:09:49
also use one of the best ID available for Java which is intell J the ultimate version of course because in community version I guess spring will not be supported so you have to go for the ultimate version and it is awesome the only thing is you have to it's not a free version but then on the other hand what if I say spring officially gives you a separate IDE it's not compulsory to use that you can still use net beans or Eclipse so the so software name is springing tool switch so if you search

00:10:15
for it you can see it will take you to the official website just click on that and you can download this spring STS for free now you'll be thinking oh we have to learn a new IDE now not exactly if you have worked on Eclipse before spring is Eclipse so it's a eclipse IDE on top of that we have a spring layer in fact if you just go here it says based on Eclipse can you see that so yes ultimately you are using Eclipse you're not using something else is just that you don't have to add a spring plugin

00:10:42
Eclipse to use it right you can simply use stts so you can download any version which you like so let me for this machine I will download the windows 64 bit because this is a Windows machine now the only thing is of course it will take some time for download so if you want to install this version you also need to have a jdk available so jdk is something which is prerequisite but then which one so the suggested one is the minimum requirement for this is jdk 8 because we'll be working on Spring

00:11:07
framework 5 for which jdk 8 is The Benchmark the reason is in Spring 5 we got reactive

programming it supports spring uh reactive programming now and reactive programming can be achieved with the help of lam expession which is only available in Java 8 but then spring framework 5 also supports Java 9 so that's great if you have Java 9 with you you can also Implement Java 9 features in Spring 5 but will not be focusing on that we'll be only focusing on Java 8 features with spring framework 5 uh so

00:11:34
once it gets downloaded you can install it you can unzip it and you can use it but how to do that that will see in the next [Music] video now once you have your SDS with you of course you have to unzip it because that's a zip version the only tip is when you unzip it make sure that you unzip with the help of a small name because if you unzip that with with a big name it may give you issue the issue which I was facing from a long time uh so make sure that you give a small name May remove that version number

00:12:02
there now once you got the unzip version you can see it says STS bundle let's double click on that now it will give you this thing which is awesome you know because when you work with Eclipse you have to download Tomcat separately that's what you do and of course Tomcat is not that heav to download but then pyot says hey when you are using SS we will give you Tom as well so it will give you pyot TC server which is pre-configured for this and then we have STS as well so let's double click on STS

00:12:28
and we got s is here and you can see that beautiful logo of it double click on this it will open just like Eclipse because behind the scene is Eclipse right it will take some time for you when you open it for the first time because you know it will a lot of files now it will ask you for the workspace name as we do in in Eclipse so here I will give a name as spring po again you can give any name doesn't matter it will simply get a folder in your document so that in which you'll be having all your

00:12:52
projects simply click on launch and it will take some time to launch it since we are doing it for the first time so fast forward you can see it is still opening and so that's my friend is STS for you yes it looks like Eclipse now I will close this dashboard of course you can see we have reference documentation available here as well simply click this and you can see this is so if you are new to any ID if you have not worked on net beans E before on the left hand side you can see we have a project section

00:13:17
because this is where you will be having all your projects in the center you have a area where you write a code and on the in the below section you can see we have console here so all your logs will be printed that if you want to print something on a console that's where you will get the output now let me create my first project now if you can see we can get multiple type of projects here we can create a spring start project which we are not going to see now that's a surprise we'll see how

00:13:41
we can you know make spring easy the second thing is we have spring Legacy project will not do that now we can create a jav project but there's one issue with this the issue is when you work with spring framework we need to add a spring dependencies right and when you say you want to add spring dependencies you have to go to Google you have to search for all the dependencies which you need and then you have to download it manually but what if I say you can do that with the help of Maven because Maven will help you here

00:14:06
so mavin is basically a build tool which will help you to get dependencies which will help you to create a package for your project which will help you to compile your codes as well but at this point let me use M only for downloading the dependencies let me click on M project okay so you can see it will say you want to create a project yes I want to get a new project now mavin has multiple types I want you to also know about Maven as I mentioned you can just go to YouTube or Google you can search

00:14:30
how Maven works but just to give you a brief using Maven you can create multiple type of projects you can create a simple stand on Project you can create a web project let me create a quick start project which will have nothing a simple project uh click on next now you can give a special name to it now we have to also mention group ID time I will mention com. Tesco as my group ID the artifact ID will say this is demo project and then click on finish now once you click on finish it will download the MAV dependency it will

00:14:58
create a project for you you with the all required dependencies but what if you want some extra dependency because if you see this project will have nothing it will not have spring dependencies it will not have hypernet of course we have not mentioned that as of now so how do you add dependencies what if you need a jarfile for spring it's actually very easy you can just open your pal. XML now mavin works with this file in which you have to mention the dependencies you can see this is where you have to mention what

00:15:24
dependency you need now if you think mer is the only option not exactly we can also achieve this with the help of griddle so you can use gridle or you can use mavin that's your choice I guess a lot of people nowadays they are moving towards griddle but again mavin also works now question arise from where you will get this dependency so you have to type it you by yourself you have to just go here now the thing is it is downloading some artifacts for us from the internet as of now let's keep it on

00:15:46
hold this is where you have to put your dependency tags but we don't know what tab we have to type right we don't know the artifact name on for the spring framework in that case let's go to the official website again now for this we will go to Google and search for mvn so we'll search for mvn repository now this is where you can download all the me dependencies so let's click on this website and this is where you will get for spring so just go here and search for spring and you have to download

00:16:14
spring context okay so not spring spring context because spring context will also have spring code uh let's click on this now you can go with any version my preferred version is always where you have maximum usage so you can see we have a usage of 419 which is which looks stable so let's go for 5.1.3 let's copy this in fact when you click on that it will automatically copy it so just go back to your application and paste you just have to specify that and if you go to my dependencies now on

00:16:40
the left hand side you can see by default we have J unit but hold on very spring if you can see it says building workspace so if you click on that it is downloading all the dependencies yes it takes

some time depend upon your internet speed unfortunately my inter speed is not that good today so it takes some time and you can see we got dependent denes which we wanted right so if you don't have to download this jar file one by one it will get downloaded automatically the moment you paste that

00:17:05
code so this is working you got your dependencies and now you can write your first application welcome back everyone my name is ready and in this video we are going to talk about spring core now spring core is the first module which we discuss from Spring framework now it's quite famous because it solved one of the biggest issue in the software world which is dependencies now I'm not talking about labes okay I'm talking about the objects think about this the Java world everything is object right in

00:08:58
know KJ that's fine you can just just pause this video and know pause the series go back to my COA videos watch some of them which is really available on this website just go there and you can learn basic Co Jaa and you can come back here that's completely fine so that's the prerequisite you should be having and that's it that's it [Music] everyone now it's time to talk about the requirement in terms of software of course right if you want to write your spring code where will write it in fact

00:09:24
the we have the simple way right you can open notepad type your code and you can run it in command line we can do that but hold on we will be building big projects right so of course if you're learning spring framework in future you will be building amazing projects and that's where if you have a good ide with you it will be an awesome thing right and that's where we have lot of Ides available you can use net beans which is everyone's favorite you can use Eclipse which is industrywide famous you can

00:09:49
also use one of the best ID available for Java which is intell J the ultimate version of course because in community version I guess spring will not be supported so you have to go for the ultimate version and it is awesome the only thing is you have to it's not a free version but then on the other hand what if I say spring officially gives you a separate IDE it's not compulsory to use that you can still use net beans or Eclipse so the so software name is springing tool switch so if you search

00:10:15
for it you can see it will take you to the official website just click on that and you can download this spring STS for free now you'll be thinking oh we have to learn a new IDE now not exactly if you have worked on Eclipse before spring is Eclipse so it's a eclipse IDE on top of that we have a spring layer in fact if you just go here it says based on Eclipse can you see that so yes ultimately you are using Eclipse you're not using something else is just that you don't have to add a spring plugin

00:10:42
Eclipse to use it right you can simply use stts so you can download any version which you like so let me for this machine I will download the windows 64 bit because this is a Windows machine now the only thing is of course it will take some time for download so if you want to install this version you also need to have a jdk available so jdk is something which is prerequisite but then which one so the suggested one is the minimum requirement for this is jdk 8 because we'll be working on Spring

00:11:07

framework 5 for which jdk 8 is The Benchmark the reason is in Spring 5 we got reactive programming it supports spring uh reactive programming now and reactive programming can be achieved with the help of lam expession which is only available in Java 8 but then spring framework 5 also supports Java 9 so that's great if you have Java 9 with you you can also Implement Java 9 features in Spring 5 but will not be focusing on that we'll be only focusing on Java 8 features with spring framework 5 uh so

00:11:34
once it gets downloaded you can install it you can unzip it and you can use it but how to do that that will see in the next [Music] video now once you have your SDS with you of course you have to unzip it because that's a zip version the only tip is when you unzip it make sure that you unzip with the help of a small name because if you unzip that with with a big name it may give you issue the issue which I was facing from a long time uh so make sure that you give a small name May remove that version number

00:12:02
there now once you got the unzip version you can see it says STS bundle let's double click on that now it will give you this thing which is awesome you know because when you work with Eclipse you have to download Tomcat separately that's what you do and of course Tomcat is not that heav to download but then pyot says hey when you are using SS we will give you Tom as well so it will give you pyot TC server which is pre-configured for this and then we have STS as well so let's double click on STS

00:12:28
and we got s is here and you can see that beautiful logo of it double click on this it will open just like Eclipse because behind the scene is Eclipse right it will take some time for you when you open it for the first time because you know it will a lot of files now it will ask you for the workspace name as we do in in Eclipse so here I will give a name as spring po again you can give any name doesn't matter it will simply get a folder in your document so that in which you'll be having all your

00:12:52
projects simply click on launch and it will take some time to launch it since we are doing it for the first time so fast forward you can see it is still opening and so that's my friend is STS for you yes it looks like Eclipse now I will close this dashboard of course you can see we have reference documentation available here as well simply click this and you can see this is so if you are new to any ID if you have not worked on net beans E before on the left hand side you can see we have a project section

00:13:17
because this is where you will be having all your projects in the center you have a area where you write a code and on the in the below section you can see we have console here so all your logs will be printed that if you want to print something on a console that's where you will get the output now let me create my first project now if you can see we can get multiple type of projects here we can create a spring start project which we are not going to see now that's a surprise we'll see how

00:13:41
we can you know make spring easy the second thing is we have spring Legacy project will not do that now we can create a jav project but there's one issue with this the issue is when you work with spring framework we need to add a spring dependencies right and when you say you want to add spring dependencies you have to go to Google you have to search for all the dependencies which you need and then you have to download it manually but what if I say you can do that with the

help of Maven because Maven will help you here

00:14:06
so mavin is basically a build tool which will help you to get dependencies which will help you to create a package for your project which will help you to compile your codes as well but at this point let me use M only for downloading the dependencies let me click on M project okay so you can see it will say you want to create a project yes I want to get a new project now mavin has multiple types I want you to also know about Maven as I mentioned you can just go to YouTube or Google you can search

00:14:30
how Maven works but just to give you a brief using Maven you can create multiple type of projects you can create a simple stand on Project you can create a web project let me create a quick start project which will have nothing a simple project uh click on next now you can give a special name to it now we have to also mention group ID time I will mention com. Tesco as my group ID the artifact ID will say this is demo project and then click on finish now once you click on finish it will download the MAV dependency it will

00:14:58
create a project for you you with the all required dependencies but what if you want some extra dependency because if you see this project will have nothing it will not have spring dependencies it will not have hypernet of course we have not mentioned that as of now so how do you add dependencies what if you need a jarfile for spring it's actually very easy you can just open your pal. XML now mavin works with this file in which you have to mention the dependencies you can see this is where you have to mention what

00:15:24
dependency you need now if you think mer is the only option not exactly we can also achieve this with the help of griddle so you can use gridle or you can use mavin that's your choice I guess a lot of people nowadays they are moving towards griddle but again mavin also works now question arise from where you will get this dependency so you have to type it you by yourself you have to just go here now the thing is it is downloading some artifacts for us from the internet as of now let's keep it on

00:15:46
hold this is where you have to put your dependency tags but we don't know what tab we have to type right we don't know the artifact name on for the spring framework in that case let's go to the official website again now for this we will go to Google and search for mvn so we'll search for mvn repository now this is where you can download all the me dependencies so let's click on this website and this is where you will get for spring so just go here and search for spring and you have to download

00:16:14
spring context okay so not spring spring context because spring context will also have spring code uh let's click on this now you can go with any version my preferred version is always where you have maximum usage so you can see we have a usage of 419 which is which looks stable so let's go for 5.1.3 let's copy this in fact when you click on that it will automatically copy it so just go back to your application and paste you just have to specify that and if you go to my dependencies now on

00:16:40
the left hand side you can see by default we have J unit but hold on very spring if you can see it

says building workspace so if you click on that it is downloading all the dependencies yes it takes some time depend upon your internet speed unfortunately my inter speed is not that good today so it takes some time and you can see we got dependent denes which we wanted right so if you don't have to download this jar file one by one it will get downloaded automatically the moment you paste that

00:17:05
code so this is working you got your dependencies and now you can write your first application welcome back everyone my name is ready and in this video we are going to talk about spring core now spring core is the first module which we discuss from Spring framework now it's quite famous because it solved one of the biggest issue in the software world which is dependencies now I'm not talking about labes okay I'm talking about the objects think about this the Java world everything is object right in

00:17:36
fact we call Java as objectoriented programming which means if you want to achieve anything in Java you have to get object and that's great right but there's only one issue with this the issue is when you work with objects we don't just create one object right you create multiple objects example uh let's say if you want to build a laptop so let's imagine you have a laptop class and which you want to specify some properties now if you want to use laptop laptop will have certain properties

00:18:02
right the first one laptop will have a size of laptop a laptop will have a weight of the laptop right every laptop varies in in their weight laptop will also have a color so we can have all those properties but to make a laptop work we need a special objects inside it example if you want to build a laptop a laptop needs a hard drive a laptop needs a ram a laptop needs a CPU right so we need all those stuff right apart from this we also need keyboard and monitor but imagine we need all those things now

00:18:28
you have two choice you can build all those objects by yourself now you will say we can do that right what we can do is we of course you'll be having a class for CPU you'll be having a class for Ram you'll be having a class for hard drive so what you will simply do you will create those objects right so you will simply say CPU opj equal to new CPU you will say hard drive whatever class name you have HDD obj equal to new HDD but we have an issue here the issue is you are creating the object right that means you

00:18:55
are responsible to destroy it that's an issue okay uh when you create the object you are responsible to destroy that the second issue we have is what if you have two classes you don't have one CPU class you have two CPU classes one is Intel class one is AMD class and now if you want to design your own laptop you have two Choice which one you will prefer and maybe you want to prefer that at run time you don't want to do that compile time you want to say hey I want a CPU but I'm not sure what object I'll be

00:19:22
creating maybe it will be Intel CPU or a AMD CPU because both these classes extends a CPU class and that's where problem starts you know that's where you want this dependencies when I say dependency I'm not talking about libraries I'm talking about the objects because your laptop will not work without this objects so this dependen is I want it to be injected in the application right so

so when I when I want to build a laptop this dependencies should be coming from outside the world

00:19:49
so maybe I will have a consultant and say hey consultant I want hard drives I want Rams I want CPUs for my machine building and that's where you don't have to make those devices you don't have to make your own CPU you don't have to make your own RAM and that's what dependency injection will give you now dependency injection has one more name to it which is ioc in fact we can interchange them uh we have ioc which is inversion of control or normally we refer that in in Spring framework as dependency injection

00:20:16
so further topics we'll be only focusing on dependency injection we'll not talk about ioc but again if someone says hey what is ioc in Spring framework you can still talk about dependency injection it will work in fact dependency injection is very famous you can do that by yourself as well you can write the entire application you can use some designed patterns to achieve that but then you have to do that everything by yourself you have to make make sure that your class is singlet turn if you if you

00:20:39
have that intention you have to get your factory classes but what if I say in Spring frame work you don't have to worry about all those things because we have a moduel which is spring core in which you simply have to mention what dependency you need spring framework will give it to you I know that looks like a magic but then it's possible but I'm not saying everything is Magic okay not everything will be happening automatically you have to do the configuration I know that's a tricky

00:21:04
part but then again we have something in future which where you will where you don't have to do everything by yourself you can reduce that configuration but but as of now if you want to achieve that if you want to achieve spring code dependency injection we have to do some configuration now example let's say we go for laptop itself so let's say if you want to build a laptop you need hard drive right so you don't have to create object of hard drive it will be done it will be injected in your application but

00:21:30
how you do that that we'll see in the next [Music] video Welcome Back everyone till this point we have seen what is frame framework and then we have also done this setup for STS now of course right spring is one of the best framework available for Java there are some other Frameworks as well example we have STS for web development or we have hibernate for database and yes we can use all those Frameworks right now Frameworks are awesome but there's only one issue when you talk about Frameworks and that

00:22:04
is we have to do a lot of configuration to achieve the thing which we want in fact even if you're building a small application we need to do a lot of configuration but think about this if you have so let's say in this world we have so many people building different projects but all this project will have something in common so all the configuration will have some configuration which is common for all the projects don't you think the basic configuration should be given by Spring framework itself uh from a long time you

00:22:32

know when spring was introduced everyone loved it but then there was still the problem with the configuration but now we have a concept of spring boot Now using spring boot spring boot basically means you know uh spring boot believes that you should focus on convention instead of focusing on configuration so it normally States convention over configuration so you have to focus on your logic don't focus on the configuration okay when I say don't focus on configuration talking about the basic configuration what if you want to

00:23:01
have something customized that's where you have to do some changes right but for basic spring will give you our spring boot will give you all the basic configuration okay now to understand that let me create a simple project on Spring boot but then don't worry we'll be also focusing on Spring framework so this video we'll talk about spring boot how do we create a project and how do we uh use some features there but then the further videos will focus only on Spring framework now here first of all let's

00:23:28
project now first of all you need STS here which is which we have already done and then we had this project as well which is demo project in the last video but now let's do something different let's say new and let's create a spring stter project as of now it's not important to understand what is this service URL stuff the only thing which is important here is the project name I will say this is my first project of course you can give any name and then I will give a group ID which is my company

00:23:52
name itself I will say com. telescope and then here the package name I want is T discod DOT first that's my project name and then package name let me click on next and here you can see it will give you so many options right now this is different okay this the project which we are building now is a spring boot project so when you say spring start a project it means spring boot it will give you so many features right based on whatever project you are building let's say if you're building a web project you

00:24:19
will be selecting web here if you want to build a project with database of course you have to search for jdbc in fact if you search for jdbc you can see we have an option here uh what if you want to work with mq you can choose that you can choose post there are so many options here what if you want to implement security and that's why we have security features as well but as of now we don't need anything and there's one more thing important which is spring boot version now of course in new

00:24:43
version we get new features as of now 2.1.2 is stable so let's go with that click on next nothing much specific here click on finish now you can see we got a project here okay is it so fast not actually you know when you're doing this for the first time it will download all the project structure and the dependencies from the internet right so it will take some time but on this machine I've already did this so I have a local copy of the Java files and if you can see if I open this we already

00:25:11
have a Java file here which is first project application. Java and the amazing thing is it already had some code in it because okay at this point we don't understand what that code is but let's keep it there you can see if I open this m dependencies we have so many dependencies here by default and that's the magic right spring boot will give you everything which you need all the basic stuff of course if you want to have some extra features we have to add those dependencies but as of now

00:25:38
it will give you everything which is required okay and then the next important file here is the spa. XML so if you have worked on MAV projects before you know what his form file is right so this is where you will add requir dependencies I'm talking about the jar files in this case we wanted to work with spring boot so you can see we have added two dependencies one for spring Boot and for testing purpose as well nothing much here let's create our first stuff now what exactly I want to create that we see in the next

00:26:08
[Music] video in the last video we have seen how to create a simple spring boot application but now let's understand how exactly we can use it so let's say we have this main file here in which we have nothing but then here I want to create the object now if you remember if you go back to the theory video which you have talked about which is dependencies or dependency injection normally what happens you know when you work on complex projects that's where you need to add some extra objects so

00:17:36
fact we call Java as objectoriented programming which means if you want to achieve anything in Java you have to get object and that's great right but there's only one issue with this the issue is when you work with objects we don't just create one object right you create multiple objects example uh let's say if you want to build a laptop so let's imagine you have a laptop class and which you want to specify some properties now if you want to use laptop laptop will have certain properties

00:18:02
right the first one laptop will have a size of laptop a laptop will have a weight of the laptop right every laptop varies in in their weight laptop will also have a color so we can have all those properties but to make a laptop work we need a special objects inside it example if you want to build a laptop a laptop needs a hard drive a laptop needs a ram a laptop needs a CPU right so we need all those stuff right apart from this we also need keyboard and monitor but imagine we need all those things now

00:18:28
you have two choice you can build all those objects by yourself now you will say we can do that right what we can do is we of course you'll be having a class for CPU you'll be having a class for Ram you'll be having a class for hard drive so what you will simply do you will create those objects right so you will simply say CPU opj equal to new CPU you will say hard drive whatever class name you have HDD obj equal to new HDD but we have an issue here the issue is you are creating the object right that means you

00:18:55
are responsible to destroy it that's an issue okay uh when you create the object you are responsible to destroy that the second issue we have is what if you have two classes you don't have one CPU class you have two CPU classes one is Intel class one is AMD class and now if you want to design your own laptop you have two Choice which one you will prefer and maybe you want to prefer that at run time you don't want to do that compile time you want to say hey I want a CPU but I'm not sure what object I'll be

00:19:22
creating maybe it will be Intel CPU or a AMD CPU because both these classes extends a CPU class

and that's where problem starts you know that's where you want this dependencies when I say dependency I'm not talking about libraries I'm talking about the objects because your laptop will not work without this objects so this dependen is I want it to be injected in the application right so so when I when I want to build a laptop this dependencies should be coming from outside the world

00:19:49
so maybe I will have a consultant and say hey consultant I want hard drives I want Rams I want CPUs for my machine building and that's where you don't have to make those devices you don't have to make your own CPU you don't have to make your own RAM and that's what dependency injection will give you now dependency injection has one more name to it which is ioc in fact we can interchange them uh we have ioc which is inversion of control or normally we refer that in in Spring framework as dependency injection

00:20:16
so further topics we'll be only focusing on dependency injection we'll not talk about ioc but again if someone says hey what is ioc in Spring framework you can still talk about dependency injection it will work in fact dependency injection is very famous you can do that by yourself as well you can write the entire application you can use some designed patterns to achieve that but then you have to do that everything by yourself you have to make make sure that your class is singlet turn if you if you

00:20:39
have that intention you have to get your factory classes but what if I say in Spring frame work you don't have to worry about all those things because we have a moduel which is spring core in which you simply have to mention what dependency you need spring framework will give it to you I know that looks like a magic but then it's possible but I'm not saying everything is Magic okay not everything will be happening automatically you have to do the configuration I know that's a tricky

00:21:04
part but then again we have something in future which where you will where you don't have to do everything by yourself you can reduce that configuration but but as of now if you want to achieve that if you want to achieve spring code dependency injection we have to do some configuration now example let's say we go for laptop itself so let's say if you want to build a laptop you need hard drive right so you don't have to create object of hard drive it will be done it will be injected in your application but

00:21:30
how you do that that we'll see in the next [Music] video Welcome Back everyone till this point we have seen what is frame framework and then we have also done this setup for STS now of course right spring is one of the best framework available for Java there are some other Frameworks as well example we have STS for web development or we have hibernate for database and yes we can use all those Frameworks right now Frameworks are awesome but there's only one issue when you talk about Frameworks and that

00:22:04
is we have to do a lot of configuration to achieve the thing which we want in fact even if you're building a small application we need to do a lot of configuration but think about this if you have so let's say in this world we have so many people building different projects but all this project will have something in common so all the configuration will have some configuration which is common for all the projects don't you think the basic configuration should be given by Spring framework

itself uh from a long time you

00:22:32
know when spring was introduced everyone loved it but then there was still the problem with the configuration but now we have a concept of spring boot Now using spring boot spring boot basically means you know uh spring boot believes that you should focus on convention instead of focusing on configuration so it normally States convention over configuration so you have to focus on your logic don't focus on the configuration okay when I say don't focus on configuration talking about the basic configuration what if you want to

00:23:01
have something customized that's where you have to do some changes right but for basic spring will give you our spring boot will give you all the basic configuration okay now to understand that let me create a simple project on Spring boot but then don't worry we'll be also focusing on Spring framework so this video we'll talk about spring boot how do we create a project and how do we uh use some features there but then the further videos will focus only on Spring framework now here first of all let's

00:23:28
project now first of all you need STS here which is which we have already done and then we had this project as well which is demo project in the last video but now let's do something different let's say new and let's create a spring stter project as of now it's not important to understand what is this service URL stuff the only thing which is important here is the project name I will say this is my first project of course you can give any name and then I will give a group ID which is my company

00:23:52
name itself I will say com. telescope and then here the package name I want is T discod DOT first that's my project name and then package name let me click on next and here you can see it will give you so many options right now this is different okay this the project which we are building now is a spring boot project so when you say spring start a project it means spring boot it will give you so many features right based on whatever project you are building let's say if you're building a web project you

00:24:19
will be selecting web here if you want to build a project with database of course you have to search for jdbc in fact if you search for jdbc you can see we have an option here uh what if you want to work with mq you can choose that you can choose post there are so many options here what if you want to implement security and that's why we have security features as well but as of now we don't need anything and there's one more thing important which is spring boot version now of course in new

00:24:43
version we get new features as of now 2.1.2 is stable so let's go with that click on next nothing much specific here click on finish now you can see we got a project here okay is it so fast not actually you know when you're doing this for the first time it will download all the project structure and the dependencies from the internet right so it will take some time but on this machine I've already did this so I have a local copy of the Java files and if you can see if I open this we already

00:25:11
have a Java file here which is first project application. Java and the amazing thing is it already had some code in it because okay at this point we don't understand what that code is but let's keep it

there you can see if I open this m dependencies we have so many dependencies here by default and that's the magic right spring boot will give you everything which you need all the basic stuff of course if you want to have some extra features we have to add those dependencies but as of now

00:25:38
it will give you everything which is required okay and then the next important file here is the spa. XML so if you have worked on MAV projects before you know what his form file is right so this is where you will add requir dependencies I'm talking about the jar files in this case we wanted to work with spring boot so you can see we have added two dependencies one for spring Boot and for testing purpose as well nothing much here let's create our first stuff now what exactly I want to create that we see in the next

00:26:08
[Music] video in the last video we have seen how to create a simple spring boot application but now let's understand how exactly we can use it so let's say we have this main file here in which we have nothing but then here I want to create the object now if you remember if you go back to the theory video which you have talked about which is dependencies or dependency injection normally what happens you know when you work on complex projects that's where you need to add some extra objects so

00:26:38
your application will need some extra objects maybe those objects needs extra objects example let's say if you build a laptop you need a hard drive and to build a hard drive you need some other parts as well so you don't have to build so as a laptop manufacturer you don't have to build hard drive you can inject it or someone else will inject the hard drive in your laptop you just have to ask for it and even you don't even have to mention what company hard drive you need you simp you can simply Say Hey I

00:27:05
want an hard drive now based on your consultant or based on your team they will say okay let's use Cate hard drive let's use maybe Hitachi hard drive that's their choice so here what I want is I can create the object of alien so I will say alien obj but hold on I'm not creating the object I'm just creating the reference first to create the reference of alien we need a alien class right unfortunately we don't have this class okay let's create one okay when I say alien I'm only talking about so

00:27:32
normally I refer alien so all the programmers for me are aliens in fact not just programmers the tech professionals are aliens because we don't think in real world right we think in virtual world so let's create the class I will simply click here and you can see I'm creating a class the class name is alien and a very simple class now in this class I want to have a simple method okay like the method name would be let's say public void cool code because normally as a programmer we

00:28:00
write code so we got code and in this I'm printing I'm coding that's it nothing much we got a alien class in which you have a method is code and then we are saying I'm coding let's go back to our application here now I want to call that method which is code of course right you can simply say obj do code but unfortunately this will not work you know why it will not work because we have not initialized this object here and that's what it's saying you can see it says the local variable obj may not

00:28:27

have been initialized now of course this OB local variable inside the main method how can we initialize it the best way we can do that is by saying new alien but that's the problem starts you are saying new which means you as a programmer you are creating the object and that's where problem starts because when you create the object you are responsible to maintain it we don't want to do that right that's we are using spring framework come on we can inject the object but who will create the object

00:28:53
for us and that's where spring says hey if you want object I will give you the object you just have to ask for it how will you ask for it it's very simple you can just go here and say hey Spring give me a bean spring will say okay I will give you a bean but tell me which class object you need do you need object of alien class do you need object of object class do you need object of added class there are so many classes available right Which object do you need and that's why you will say okay you're

00:29:20
getting confused that means I have to mention so I will simply say alien. class so you're saying hey Spring give me the object of alien class okay spring will say okay but then there's one issue right in Java if you want to call a method we have to use the object of course right you can't call the method without the object so here what I will do is I will simply say which class object do you need so this get bean belongs to an interface which is application context so let me repeat get

00:29:48
bean method belongs to an interface named AS application context so if you want to call get bean we need object of application context the amazing thing is if you click on this run method so you can see we have a line here which is Spring application. Run which takes the class name itself if you click on this run you can see we got documentation of course so how do you get the documentation you just have to press control and click on the method name which you want so you can see this run method returns an object of configurable

00:30:18
application context this class or this interface is it interface yes this interface extends application context so that means this run is actually giving you the object of application context so we can simply say application context let me import the package I will say context equal to we got the object now once you got the context object I can simply come here and you can say context. getb our job is done will this work let's try let's go back here and say right click run as spring boot app I

00:30:52
hope this will work let's try you normally learn by erors right so let's let's see and and we got error the error it says no qualifying being of type com. telescope. first. alen but we have a class there right it's just that see who is responsible to create the object spring framework right so spring responsibility to create the object of alien and you can mention that to Spring by saying add component So when you say add component on top of your class it simply means spring framework will know....give important and teach me like this only in very very simple

Okay, Samiksha, let’s do this like Navin sir but even slower and simpler 😄
I’ll not repeat everything from the video, I’ll **pull out only the important parts** and explain in **very easy language with small examples.**

1️⃣ What is Spring (in baby language)

Think like this:

- You have **Java** = language.
- You want to build **big real-world projects** = websites, APIs, microservices.
- Doing everything with only core Java = **too much manual work** (like building a house alone).

👉 **Spring = a helper framework** that:

- Manages objects for you (beans)
- Connects layers (web, service, database)
- Handles security, transactions, etc.

So instead of you writing everything from scratch,
Spring gives you ready-made tools and you just plug your business logic.

2️⃣ Old World vs Spring World (why Spring came)

Earlier in Java:

- For web: **Servlet + JSP**, later **Struts**
- For big enterprise apps: **EJB**
- For DB: **JPA / Hibernate**

Problems:

- Heavy configuration
- Complex classes (EJB)
- Repeated setup

Spring said:

> "I'll let you write **simple POJO classes** and I'll add all magic around them."

🧩 **POJO** = Plain Old Java Object
Just a normal Java class (no special parent class, no framework inheritance).

Example:

```java
public class Laptop {
    private int size;
    private int ram;
    public void start() {
        System.out.println("Laptop started");
    }
}
```

This is a POJO. Spring loves such classes.

3️⃣ Spring is not one thing, it's many modules

Important idea from the video:

> "Spring is a collection of modules"

Some important ones:

- **Spring Core** → Dependency Injection (creating & giving objects)
- **Spring MVC** → Web (controllers, views)
- **Spring Web / REST** → REST APIs
- **Spring Data JPA** → Database with JPA/Hibernate
- **Spring Security** → Security (login, roles)
- **Spring AOP** → Cross-cutting (logging, transactions)
- **Spring Boot** → Makes using Spring super easy (auto config)

You **never use "everything" at once**.
You pick modules based on your project.

4️⃣ Prerequisites (what you *should* know)

From the video:

- ✅ Core Java (Oops, exceptions, threads, loops)
- ✅ JDBC (how Java talks to DB)
- ✅ Servlets & JSP (**basic idea** of HTTP request/response)
- ✅ ORM (Hibernate/JPA) – at least concept level

Don't panic if you don't know all deeply.
Just keep revising slowly. You'll understand more as you see code.

5️⃣ Tools: IDE + Maven + STS

You can write Java in Notepad + cmd, but for Spring projects that's painful.

So:

- **IDEs**: Eclipse, IntelliJ, NetBeans
- **Spring Tool Suite (STS)** = Eclipse + Spring plugins already added
  (so you don't manually setup Spring support)

Also, we use:

Maven (very important 🚨)

Maven helps you with:

- 📦 **Dependencies** = jar files (Spring, Hibernate, etc.)
- 🏗️ Build = compile, create jar/war
- ⚙️ Project structure

Maven uses **`pom.xml`**.
In **`pom.xml`**, you write:

```xml
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.1.3.RELEASE</version>
</dependency>
```

Then Maven says:

> “Okay, I’ll download this jar from the internet and add it to your project.”

You get these snippets from **mvnrepository.com** by searching.

Important point:
You **don’t** download jar files manually. Maven handles it.

6️⃣ Spring Boot – Why it exists?

Problem with normal Spring:

- You need many XML config or Java config.
- You have to:
    - Configure Tomcat
    - Configure dispatcher servlet
    - Configure data source
    - Configure component scan
    - Add a lot of boilerplate

Spring Boot says:

> “Most projects have **common configuration**, so I will automatically do the default setup for you.”

This idea is called:

> ✅ **Convention over configuration**
> (Follow standard naming & structure → Spring Boot will auto-configure a lot.)

Spring Boot gives you:

- Embedded Tomcat (no need to install server separately)
- Auto-configuration
- Starter dependencies (like **`spring-boot-starter-web`**)
- Easy project creation (Spring Starter / Spring Initializr)

7️⃣ The Alien Example – Understanding Spring Core + Boot

In the video, he creates:

- A **Spring Boot project**
- A simple **class called `Alien`**
- And uses **Spring to create the object**, not **`new`**.

Step 1: The `Alien` class

```java
public class Alien {
    public void code() {
        System.out.println("I'm coding");
    }
}
```

Normal Java way:

```java
Alien obj = new Alien();
obj.code();
```

But this has issues:

- You are controlling object creation (**`new`**).
- In big apps, many objects depend on other objects.
- Hard to manage lifecycle (who creates, when destroyed, etc.).
- Hard to switch implementation (IntelCpu vs AmdCpu etc.).

Step 2: Let Spring create objects (beans)

We want to say:

> "Spring, please create and manage the **`Alien`** object for me.
> I will just **ask** you whenever I need it."

For that, we do:

```java
@Component
public class Alien {
    public void code() {
        System.out.println("I'm coding");
    }
}
```

`@Component` =
💬 "Spring, this is a bean. Please create an object of this class and manage it."

Now `Alien` is no longer just your random class;
it is a **Spring bean**.

8️⃣ `SpringApplication.run(...)` and `ApplicationContext`

In your main class:

```java
@SpringBootApplication
public class FirstProjectApplication {

    public static void main(String[] args) {
        ApplicationContext context =
                SpringApplication.run(FirstProjectApplication.class, args);

        Alien obj = context.getBean(Alien.class);
        obj.code();
    }
}
```

Let's break it s-l-o-w-l-y 👇

🔹 `@SpringBootApplication`

This is like a combo of:

- `@Configuration`
- `@EnableAutoConfiguration`
- `@ComponentScan`

You can remember it as:

> "This is a Spring Boot main class. Start Spring from here and scan my packages for beans."

🔹 `SpringApplication.run(...)`

This line:

```java
ApplicationContext context =
    SpringApplication.run(FirstProjectApplication.class, args);
```

does 3 big things:

1. Starts Spring Boot application.
2. Creates the **Spring container** (called `ApplicationContext`).
3. Scans your package, finds `@Component` classes, creates their objects (beans).

You can imagine:

> `ApplicationContext` = **a box that holds all Spring-managed objects**.

Alien object is inside this box because of `@Component`.

🔹 `context.getBean(Alien.class)`

Now instead of:

```java
Alien obj = new Alien();
```

we do:

```java
Alien obj = context.getBean(Alien.class);
```

Meaning:

> "Hey Spring container, give me the `Alien` object you are managing."

Spring returns the bean, and then:

```java
obj.code();
```

prints:

```text
I'm coding
```

🚨 Important error from video:

They got:

> `No qualifying bean of type 'com.telusko.first.Alien'`

Why?

Because initially **they forgot** to add `@Component` on `Alien` class.

Spring was like:

> “You are asking for Alien bean, but I don’t know this class.
> You never told me to manage it.”

Once we add:

```java
@Component
public class Alien { ... }
```

Spring will:

- See this during startup
- Create an object of **`Alien`**
- Keep it inside **`ApplicationContext`**
- Then **`getBean(Alien.class)`** will work ✅

9️⃣ Key concepts you should remember (from this whole chunk)

Let me highlight the **most important points** from what you pasted:

1. **Spring = Framework** to build Java applications (web, DB, security, etc.) easily.
2. Spring is **modular**: core, MVC, data, security, AOP, boot, etc.
3. **POJOs** are simple Java classes. Spring uses POJOs heavily.
4. **Dependency Injection (DI)** = Spring creates and injects object for you.
5. **IoC container / ApplicationContext** =
   Spring’s object manager (box containing all beans).
6. **Maven** is used to manage dependencies (pom.xml).
7. **Spring Boot**:
    - Auto-configures many things
    - Gives embedded server
    - Follows convention-over-configuration
8. **`@Component`** tells Spring: "manage this class as a bean."
9. **`SpringApplication.run(...)`** returns **`ApplicationContext`**.
10. **`context.getBean(Alien.class)`** gets bean instead of **`new Alien()`**.

If you want, in the **next step** we can:

- Draw a **small diagram** of:
    - ApplicationContext
    - Bean creation
    - How **`@ComponentScan`** works

- Then move to **field injection / constructor injection** with another simple example (like Laptop–CPU–RAM).

You don’t need to say much — just tell me:

> “Next: explain dependency injection with code”

and I’ll take that as the next topic and go in the same very simple style.

okay now it says add component which means I'm responsible to create the object of alen now it should work let's go back and let's relaunch the application and it worked and you see that we got the output as I am coding it's that simple right and that's the beauty about spring framework you are not responsible to create the [Music] object so once we have done with alien think about this when you say a alien can code so I'm talking about programmers here so when you can code of

00:31:55
course you need a laptop right or desktop so if you want to code something of course we we go in multiple steps we have to compile our code and that's where you need object of laptop so let's create one so we say laptop laptop or laptop lab here I just want to create object of laptop if you want to make it work and here instead of saying I'm coding I can simply say lap. compile of course right when you code you have to compile your code as well so I will say compile now what is wrong with this code

00:32:24
there are multiple things which we are doing wrong first of all we don't have this class class named as laptop and second of all okay we we don't have a class of course we don't have a method as well so let's create this class of course you can write click here you can create a laptop class let me just use a shortcut for IDs let's click here and say hey create a laptop class for me and you can see it will create a laptop class for you that's a very simple class go back to alien you can see we also

00:32:48
have issue with the compile so I will click here and I will say create a compile method and job is done in this compile I will print of course you can write any statement here let's say compiling so for us it's not important what is happening inside this method what is important is how will this alien can get the object of laptop so of course right if you want to make it work you have to instantiate laptop as you can see we are not doing it and the moment you run this application now we

00:33:15
are saying code of course code will call uh this method and then it will call compile but to call compile first we have to instantiate laptop let me just run this application as it is I will say run as spring boot app app and now you can see we got okay it's asking for permission again and you can see we got an exception which is null Point exception in this case of course right this object here is not instantiated so of course you can use new a laptop here but then that's the idea right we don't

00:33:43
want to create the object by ourself we are not done for the alien as well and that's where we can use a special annotation here which is named as autowire so when we use autowire it simply

means that this object this laptop object will be created by springing oh hold on we're still missing something we have to mention that component here as well right you have to say add component so you have to make sure all the classes has ADD component on top of it the same thing we have done for alien

00:34:13
now what will happen is the moment of this application of course we have a detailed explanation for this in the next module uh this is just the introduction for spring boot how it can be awesome so you can see the moment you load this application it will give you two objects the object of alien the object of laptop right and in the alien class we are asking for the laptop object so it will check hey do we have object of laptop available in the memory and the answer is yes so it will give you the

00:34:39
laptop object here and that's the beauty now since you are not creating the object it will be managed by Spring framework and trust me it will give you a lot of advantage in making big projects as of now yes it doesn't make sense because we have a very small code but hold on now the more we learn about spring frame framework with different modules different projects this will surely make sense so that's about autowire okay so you can see oh we have not run this code let's do that so we'll

00:35:06
say right click run as spring boot app and you can see we got the output it says compiling that means we are able to call this code method and code is able to call this compile without any issue let me repeat we are not creating the object of laptop it is managed by Spring framework so that's it from this video in the next module we'll talk about the fancy Stu I mean the a fancy it simply means what is happening behind the [Music] scene welcome back everyone till this point we have seen some amazing stuff in

00:35:39
spring boot right we have seen how to create a project in Spring boot we have seen how to use dependency injection there and then the important concept Auto wire but then still we have not understood how spring is doing everything behind the scene and that's what we have to understand now question arise how will you understand what is happening behind the scene because in Spring boot you cannot do anything else right because everything is done by Spring Boot and that's why next few videos will not be based on Spring boot

00:26:38
your application will need some extra objects maybe those objects needs extra objects example let's say if you build a laptop you need a hard drive and to build a hard drive you need some other parts as well so you don't have to build so as a laptop manufacturer you don't have to build hard drive you can inject it or someone else will inject the hard drive in your laptop you just have to ask for it and even you don't even have to mention what company hard drive you need you simp you can simply Say Hey I

00:27:05
want an hard drive now based on your consultant or based on your team they will say okay let's use Cate hard drive let's use maybe Hitachi hard drive that's their choice so here what I want is I can create the object of alien so I will say alien obj but hold on I'm not creating the object I'm just creating the reference first to create the reference of alien we need a alien class right unfortunately we don't have this class okay let's create one okay when I say alien I'm only talking about so

00:27:32
normally I refer alien so all the programmers for me are aliens in fact not just programmers the tech professionals are aliens because we don't think in real world right we think in virtual world so let's create the class I will simply click here and you can see I'm creating a class the class name is alien and a very simple class now in this class I want to have a simple method okay like the method name would be let's say public void cool code because normally as a programmer we

00:28:00
write code so we got code and in this I'm printing I'm coding that's it nothing much we got a alien class in which you have a method is code and then we are saying I'm coding let's go back to our application here now I want to call that method which is code of course right you can simply say obj do code but unfortunately this will not work you know why it will not work because we have not initialized this object here and that's what it's saying you can see it says the local variable obj may not

00:28:27
have been initialized now of course this OB local variable inside the main method how can we initialize it the best way we can do that is by saying new alien but that's the problem starts you are saying new which means you as a programmer you are creating the object and that's where problem starts because when you create the object you are responsible to maintain it we don't want to do that right that's we are using spring framework come on we can inject the object but who will create the object

00:28:53
for us and that's where spring says hey if you want object I will give you the object you just have to ask for it how will you ask for it it's very simple you can just go here and say hey Spring give me a bean spring will say okay I will give you a bean but tell me which class object you need do you need object of alien class do you need object of object class do you need object of added class there are so many classes available right Which object do you need and that's why you will say okay you're

00:29:20
getting confused that means I have to mention so I will simply say alien. class so you're saying hey Spring give me the object of alien class okay spring will say okay but then there's one issue right in Java if you want to call a method we have to use the object of course right you can't call the method without the object so here what I will do is I will simply say which class object do you need so this get bean belongs to an interface which is application context so let me repeat get

00:29:48
bean method belongs to an interface named AS application context so if you want to call get bean we need object of application context the amazing thing is if you click on this run method so you can see we have a line here which is Spring application. Run which takes the class name itself if you click on this run you can see we got documentation of course so how do you get the documentation you just have to press control and click on the method name which you want so you can see this run method returns an object of configurable

00:30:18
application context this class or this interface is it interface yes this interface extends application context so that means this run is actually giving you the object of application context so we can simply say application context let me import the package I will say context equal to we got the object now once you got the context object I can simply come here and you can say context. getb

our job is done will this work let's try let's go back here and say right click run as spring boot app I

00:30:52
hope this will work let's try you normally learn by erors right so let's let's see and and we got error the error it says no qualifying being of type com. telescope. first. alen but we have a class there right it's just that see who is responsible to create the object spring framework right so spring responsibility to create the object of alien and you can mention that to Spring by saying add component So when you say add component on top of your class it simply means spring framework will know

00:31:22
okay now it says add component which means I'm responsible to create the object of alen now it should work let's go back and let's relaunch the application and it worked and you see that we got the output as I am coding it's that simple right and that's the beauty about spring framework you are not responsible to create the [Music] object so once we have done with alien think about this when you say a alien can code so I'm talking about programmers here so when you can code of

00:31:55
course you need a laptop right or desktop so if you want to code something of course we we go in multiple steps we have to compile our code and that's where you need object of laptop so let's create one so we say laptop laptop or laptop lab here I just want to create object of laptop if you want to make it work and here instead of saying I'm coding I can simply say lap. compile of course right when you code you have to compile your code as well so I will say compile now what is wrong with this code

00:32:24
there are multiple things which we are doing wrong first of all we don't have this class class named as laptop and second of all okay we we don't have a class of course we don't have a method as well so let's create this class of course you can write click here you can create a laptop class let me just use a shortcut for IDs let's click here and say hey create a laptop class for me and you can see it will create a laptop class for you that's a very simple class go back to alien you can see we also

00:32:48
have issue with the compile so I will click here and I will say create a compile method and job is done in this compile I will print of course you can write any statement here let's say compiling so for us it's not important what is happening inside this method what is important is how will this alien can get the object of laptop so of course right if you want to make it work you have to instantiate laptop as you can see we are not doing it and the moment you run this application now we

00:33:15
are saying code of course code will call uh this method and then it will call compile but to call compile first we have to instantiate laptop let me just run this application as it is I will say run as spring boot app app and now you can see we got okay it's asking for permission again and you can see we got an exception which is null Point exception in this case of course right this object here is not instantiated so of course you can use new a laptop here but then that's the idea right we don't

00:33:43
want to create the object by ourself we are not done for the alien as well and that's where we can use a special annotation here which is named as autowire so when we use autowire it simply means that this object this laptop object will be created by springing oh hold on we're still missing

something we have to mention that component here as well right you have to say add component so you have to make sure all the classes has ADD component on top of it the same thing we have done for alien

00:34:13
now what will happen is the moment of this application of course we have a detailed explanation for this in the next module uh this is just the introduction for spring boot how it can be awesome so you can see the moment you load this application it will give you two objects the object of alien the object of laptop right and in the alien class we are asking for the laptop object so it will check hey do we have object of laptop available in the memory and the answer is yes so it will give you the

00:34:39
laptop object here and that's the beauty now since you are not creating the object it will be managed by Spring framework and trust me it will give you a lot of advantage in making big projects as of now yes it doesn't make sense because we have a very small code but hold on now the more we learn about spring frame framework with different modules different projects this will surely make sense so that's about autowire okay so you can see oh we have not run this code let's do that so we'll

00:35:06
say right click run as spring boot app and you can see we got the output it says compiling that means we are able to call this code method and code is able to call this compile without any issue let me repeat we are not creating the object of laptop it is managed by Spring framework so that's it from this video in the next module we'll talk about the fancy Stu I mean the a fancy it simply means what is happening behind the [Music] scene welcome back everyone till this point we have seen some amazing stuff in

00:35:39
spring boot right we have seen how to create a project in Spring boot we have seen how to use dependency injection there and then the important concept Auto wire but then still we have not understood how spring is doing everything behind the scene and that's what we have to understand now question arise how will you understand what is happening behind the scene because in Spring boot you cannot do anything else right because everything is done by Spring Boot and that's why next few videos will not be based on Spring boot

00:36:07
it will be based on simple spring framework now when you say simple spring framework that's where you are responsible to specify everything what is the scope of your bin is it single turn or prototype again we'll talk about that in detail don't worry but we have so many things which is happening behind the scene that's what we are going to see now so let's get started so step one we will be creating a project in Spring okay not a spring boot project a simple spring project but then question arise

00:36:35
how will you manage dependencies I'm okay I'm not talking about object dependencies I'm talking about jar files and that's where we can use MAV right so let's create a project here I will create a MAV project and we'll name this project as okay let's say next I'm not making a web application I'm simply making code application audio what do you say a standalone software so we select quick start here click on next this is where you have to mention a group ID so let's say com. telescope and

00:37:04
the artifact ID will be spring demo of course you can go find your name it doesn't matter right and click on finish now once you got your project now this is not a spring project okay the name is only spring but then it's a normal Maven project what we want is we want to have spring features here but why so to understand that let's go back to app and this is where I want to create an object okay first of all Which object if you remember in the the spring boot videos we have talked about the class as alien

00:37:33
let's stick to that here so let me remove this hello world oh first of all let me just verify if everything is working that's what we do right when you create a new project we just have to verify if the thing which you are working with is working so let's try so let's right click here and say run as Java application you should see some output in the console and it says hello world that quits at least this is working so what next let's create an object object of alien the way you do

00:37:58
that is by going here and say Hey I want to say alien object so I will say alien obj but unfortunately we don't have this class right that's what it's saying here we don't have that class as alien so let's create one so of course you can just go there and you can create a separate class as alien otherwise you can simply use shortcuts here so let's click here and say hey create a class for me and now we'll be getting a alien class click on finish and you can see we got a very simple alien class now this

00:38:27
is where I want a very simple method which is let's say public void show or maybe public void code the same method which we have used earlier and here we'll print I'm coding nothing fancy we got a simple alien and then we got a method which is code and we are printing I'm coding okay so the idea is I want to instantiate this alien first so I will simply say equal to new alien oh hold on that's what we don't want to do right we don't want to instantiate this object by

00:38:54
ourself that's where spring comes into picture so what spring will do now now spring provides you an amazing feature of dependency injection or you can call it as ioc which is inversion of control where you don't have to create the object it will be done by Spring framework okay so let's ask spring framework here so we'll say hey Spring give me the object of alien okay we have used that method name right now if you remember we have a very special method here which is get bean now get bean is

00:39:23
responsible to give you the object but we have a twist here the twist is get bean belongs to some interface right that's the first thing second So when you say it belongs to some interface it should belong to some libraries right of course in code Java or the jdk which we normally use here we don't have this as a in class or inil method right inside jvm or inside J so we have to get it from somewhere and that's where we have to add dependencies for spring framework if you go to your M dependencies you can

00:39:54
see we don't have any spring dependency here so so that should be our first step we have to add dependency and to achieve that let's go to browser so we will go to m repository this is where you will get all dependencies right and here we will be searching for spring context now when you

search for spring context you can see we got it here let's click on that now you can see we have so many versions the latest version is 5.1.4 now depend upon when you watch this video if you're watching this in Jan Feb of

00:40:22
course you'll be working on 5.1.4 uh if you're watching this later of you'll be having some new versions but you know every minor version there will not be any big updates so we can select any of this in fact I prefer the dependencies which have more usage you can see we have 504 here so let's pick up 5.1.3 okay let's copy this code from here go back and oh where to paste it and if you are working on MAV before you know we have to just go to pal. XML file and this is where we have to add a

00:40:52
dependency so simply paste it and the moment you say save now if you're doing this for the first time it will take some time because all these dependencies will be coming from the M repository the remote one uh luckily on this laptop I've already used it so we have a local copy of it and that's why it took around 1 second to do that okay so once we got dependency let's use them where to use it go back to your app and this is where you'll be using a interface in which you have this method which is get bean okay

00:41:20
now was interface name okay you guess it right we have a concept of application context but then before application cont text came into picture we had something else think about this if you want to make a car you have to open a car factory if you want to make a mobile phone you have to make a mobile phone factory if you want to make a bean you need a you guess it right it's bean factory so we have an interface named as bean factory let's simply say control space here and you can see this bean

00:41:48
factory interface belongs to Spring framework Library so let's click that let's instantiate it we'll simply say Factory equal to so we'll say bin factory factory equal to oh we have a problem here bin Factory is a interface uh we don't have a class so let's instantiate it with the help of a class and the class name or the class which implements bean factory is XML bean factory that's it you just have to mention that so you have to mention bean factory factory equal to new XML bean

00:42:17
factory and you can see we got a strike there now this strike simply means that we should not be using this class here because this is dicated of course we have some new features you got some new classes so you should not be using this but time when since we are into learning State we'll be learning everything from start right so this is the first class which came into picture to implement bean factory time just ignore that dicated strike there okay now we have a issue the issue is when you say get bean

00:42:44
so let's say if I say Factory here I would simply say factory. getbean we still have an issue here the issue is how do you know which class object do you need so that's what you have to specify in this bracket I will simply say hey I want object of alien but see spring framework is a framework right it's not a Superman it's not a magician to know everything right so when you say alien spring should know what this alien means and that's where we have to do some configuration of

00:43:13

course right whenever you work with Frameworks there is a part of configuration that's one of the reason why people love spring boot right we have less configuration there but since we are not using spring boot we are using normal spring framework we have to configure it but how there are different ways you can configure this there's a XML based configuration we have a Java based configuration or we can also use annotations time being since we are into learning State let's use XML that means

00:43:38
if you want to customize it if you want to specify configuration we have to create XML files and then you have to specify that file here but then unfortunately we cannot simply pass a file name here if you want to specify the file we have to create one more object here which is new file system resource is it there file system resource yes and in this you have to mention the file name so file name is we'll be creating that file I will simply name it as spring.xml of course you can go with any name but let's go

00:44:08
with this okay once we got this Factory object once you have specified hey I want an alien but still we are getting eror see it's not because of we don't have that file it's just that when you say get B it will give you object right example if you see it says it will give you object we want object of alien in that case in this bracket I simply specify alien that's it you can see there's no error yes we have some warning but that's fine let's right click here at least let's see see of

00:44:37
course this will not work because we don't have spring.xml file as of now let's see the error as well so I will simply say run as Java application yes I'm expecting error here and we got an error and that's why it is saying file not found exception which is it's not able to find spring.xml okay let's created then so let's right click on your project and I will say new and this is where you'll be saying Hey I want to create a XML file that's one way of doing it otherwise you can simply say

00:45:07
see the problem is when you say XML we have to add the DDD as well if you don't want to do that you can simply say new there's the option of spring Bean configuration file and that's why we are using STS here you could have done the same thing in Eclipse as well but in Eclipse you don't get this option which is spring Bean configuration file let's click here and let's mention the file name the file name here would be spring.xml the same name which you have mentioned in the code let's click on

00:36:07
it will be based on simple spring framework now when you say simple spring framework that's where you are responsible to specify everything what is the scope of your bin is it single turn or prototype again we'll talk about that in detail don't worry but we have so many things which is happening behind the scene that's what we are going to see now so let's get started so step one we will be creating a project in Spring okay not a spring boot project a simple spring project but then question arise

00:36:35
how will you manage dependencies I'm okay I'm not talking about object dependencies I'm talking about jar files and that's where we can use MAV right so let's create a project here I will create a MAV project and we'll name this project as okay let's say next I'm not making a web application I'm simply making code application audio what do you say a standalone software so we select quick start here click on next this is where you have to mention a group ID so let's say com. telescope

and

00:37:04
the artifact ID will be spring demo of course you can go find your name it doesn't matter right and click on finish now once you got your project now this is not a spring project okay the name is only spring but then it's a normal Maven project what we want is we want to have spring features here but why so to understand that let's go back to app and this is where I want to create an object okay first of all Which object if you remember in the the spring boot videos we have talked about the class as alien

00:37:33
let's stick to that here so let me remove this hello world oh first of all let me just verify if everything is working that's what we do right when you create a new project we just have to verify if the thing which you are working with is working so let's try so let's right click here and say run as Java application you should see some output in the console and it says hello world that quits at least this is working so what next let's create an object object of alien the way you do

00:37:58
that is by going here and say Hey I want to say alien object so I will say alien obj but unfortunately we don't have this class right that's what it's saying here we don't have that class as alien so let's create one so of course you can just go there and you can create a separate class as alien otherwise you can simply use shortcuts here so let's click here and say hey create a class for me and now we'll be getting a alien class click on finish and you can see we got a very simple alien class now this

00:38:27
is where I want a very simple method which is let's say public void show or maybe public void code the same method which we have used earlier and here we'll print I'm coding nothing fancy we got a simple alien and then we got a method which is code and we are printing I'm coding okay so the idea is I want to instantiate this alien first so I will simply say equal to new alien oh hold on that's what we don't want to do right we don't want to instantiate this object by

00:38:54
ourself that's where spring comes into picture so what spring will do now now spring provides you an amazing feature of dependency injection or you can call it as ioc which is inversion of control where you don't have to create the object it will be done by Spring framework okay so let's ask spring framework here so we'll say hey Spring give me the object of alien okay we have used that method name right now if you remember we have a very special method here which is get bean now get bean is

00:39:23
responsible to give you the object but we have a twist here the twist is get bean belongs to some interface right that's the first thing second So when you say it belongs to some interface it should belong to some libraries right of course in code Java or the jdk which we normally use here we don't have this as a in class or inil method right inside jvm or inside J so we have to get it from somewhere and that's where we have to add dependencies for spring framework if you go to your M dependencies you can

00:39:54
see we don't have any spring dependency here so so that should be our first step we have to add dependency and to achieve that let's go to browser so we will go to m repository this is where you

will get all dependencies right and here we will be searching for spring context now when you search for spring context you can see we got it here let's click on that now you can see we have so many versions the latest version is 5.1.4 now depend upon when you watch this video if you're watching this in Jan Feb of

00:40:22
course you'll be working on 5.1.4 uh if you're watching this later of you'll be having some new versions but you know every minor version there will not be any big updates so we can select any of this in fact I prefer the dependencies which have more usage you can see we have 504 here so let's pick up 5.1.3 okay let's copy this code from here go back and oh where to paste it and if you are working on MAV before you know we have to just go to pal. XML file and this is where we have to add a

00:40:52
dependency so simply paste it and the moment you say save now if you're doing this for the first time it will take some time because all these dependencies will be coming from the M repository the remote one uh luckily on this laptop I've already used it so we have a local copy of it and that's why it took around 1 second to do that okay so once we got dependency let's use them where to use it go back to your app and this is where you'll be using a interface in which you have this method which is get bean okay

00:41:20
now was interface name okay you guess it right we have a concept of application context but then before application cont text came into picture we had something else think about this if you want to make a car you have to open a car factory if you want to make a mobile phone you have to make a mobile phone factory if you want to make a bean you need a you guess it right it's bean factory so we have an interface named as bean factory let's simply say control space here and you can see this bean

00:41:48
factory interface belongs to Spring framework Library so let's click that let's instantiate it we'll simply say Factory equal to so we'll say bin factory factory equal to oh we have a problem here bin Factory is a interface uh we don't have a class so let's instantiate it with the help of a class and the class name or the class which implements bean factory is XML bean factory that's it you just have to mention that so you have to mention bean factory factory equal to new XML bean

00:42:17
factory and you can see we got a strike there now this strike simply means that we should not be using this class here because this is dicated of course we have some new features you got some new classes so you should not be using this but time when since we are into learning State we'll be learning everything from start right so this is the first class which came into picture to implement bean factory time just ignore that dicated strike there okay now we have a issue the issue is when you say get bean

00:42:44
so let's say if I say Factory here I would simply say factory. getbean we still have an issue here the issue is how do you know which class object do you need so that's what you have to specify in this bracket I will simply say hey I want object of alien but see spring framework is a framework right it's not a Superman it's not a magician to know everything right so when you say alien spring should know what this alien means and that's where we have to do some configuration of

00:43:13
course right whenever you work with Frameworks there is a part of configuration that's one of the reason why people love spring boot right we have less configuration there but since we are not using spring boot we are using normal spring framework we have to configure it but how there are different ways you can configure this there's a XML based configuration we have a Java based configuration or we can also use annotations time being since we are into learning State let's use XML that means

00:43:38
if you want to customize it if you want to specify configuration we have to create XML files and then you have to specify that file here but then unfortunately we cannot simply pass a file name here if you want to specify the file we have to create one more object here which is new file system resource is it there file system resource yes and in this you have to mention the file name so file name is we'll be creating that file I will simply name it as spring.xml of course you can go with any name but let's go

00:44:08
with this okay once we got this Factory object once you have specified hey I want an alien but still we are getting eror see it's not because of we don't have that file it's just that when you say get B it will give you object right example if you see it says it will give you object we want object of alien in that case in this bracket I simply specify alien that's it you can see there's no error yes we have some warning but that's fine let's right click here at least let's see see of

00:44:37
course this will not work because we don't have spring.xml file as of now let's see the error as well so I will simply say run as Java application yes I'm expecting error here and we got an error and that's why it is saying file not found exception which is it's not able to find spring.xml okay let's created then so let's right click on your project and I will say new and this is where you'll be saying Hey I want to create a XML file that's one way of doing it otherwise you can simply say

00:45:07
see the problem is when you say XML we have to add the DDD as well if you don't want to do that you can simply say new there's the option of spring Bean configuration file and that's why we are using STS here you could have done the same thing in Eclipse as well but in Eclipse you don't get this option which is spring Bean configuration file let's click here and let's mention the file name the file name here would be spring.xml the same name which you have mentioned in the code let's click on

00:45:34
enter and we got our XML file this is what we wanted right and you can see everything is mentioned let's enter say enter here this is where you have to say hey see if you go back to your app it says find I mean get bean of alien so we just have to mention hey Spring framework whenever a user says alien give the object of alien class now alien class belongs to a package which is com. teliscope do spring demo. alen whenever in a code says alien you have to create object of alien I mean this alien the

00:46:08
class alien but we cannot simply use this syntax right we have to use syntax of spring and that's where spring says hey if you want to specify this create a bean as let's say Bean every Bean will have an ID so we'll say ID equal to and ID is alien so whatever ID you want whatever name you want to use in the code which is alien in this case mention that and also mention the class of which

you want an object I simply say I want object of com in fact we can use alien because it will give you the class as it

00:46:37
is say control space you can see it will autoc complete it and that's it that's the only thing you have to mention it's Beauty right you just have to configure it by saying hey when a user says alien give the object of alien go back to your code and now let's run this application so right click say run as Java application and oh no error but we have not also got the output the reason is we are not calling Cod so you'll be saying obj do code because as a alien as a programmer you have to code right so

00:47:09
let's run this application now I hope it will work and it work can you see that we got I am coding and that's it it's so simple so what we have done we are trying to use spring framework to get the object as you can see the object of alien is not created by you it's created by Spring frame workk I know for this example it doesn't make sense because to create one object we are doing so much of stuff right but think about the complex of application what if you building an application where you have

00:47:36
maybe 15 20 classes and every class will have let's say two three objects you don't have to manage those objects right you to manage by Spring framework but how do we do that that we see in the upcoming [Music] videos so to this point we have seen how to create a simple spring project where you can introduce a concept of Factory which will give you the object right it works but still we have an issue here the issue is you can see the code we are getting uh it says deprecated right so this class XML bin Factory is deprecated

00:48:08
we should not be using that of course you can search for some other class which implements B Factory and which will make it work without depcat of course there's one more thing which is important here when spring came into picture in 2003 it was basically for dependency injection right that was the main intention but later on they added some new features right every time you go to Spring website you will see some new projects and of course for new features you need some new methods and those methods are not available in bin

00:48:35
Factory so instead of using bin Factory we can use amazing thing which is we have a concept of application context in fact if we have seen that in Spring boot right we have this interface which is application context we simply say Factory again you can use different name but then question arise how to instantiate this because application context interface right so so we have a class which named as class path XML let's say control space so we have a class which says class path XML application context let's use that now

00:49:06
this class implements application context example if I click on click here you can see it's a class which extends abstract XML application context if I click on that it says extends abstract refreshable config application context if I click on that somewh if you go up and up and up you will find everything extends from application context are we getting there today yes we got it so you can see we have the whole hierarchy there and unfortunately by doing that I have opened so many classes here let me

00:49:35
just close them quickly okay let's get back here so instead of using B Factory now we are using application context the amazing thing is here itself you can mention spring.xml you don't have to mention new file system resource or something like that you can simply mention the thing and you can see we are not getting any dicated now because we will be using application context a lot so now let's right click and say run as Java application we have not changed much we have only changed from bin

00:50:01
Factory to appliation context let's click on run and we got an error it says file not found exception that's weird we have that file here right the problem is since you have mentioned class path right it will look for this file in class path so you have to move this file in a class path in a Java folder of course let's keep it there now and let's relaunch the application and now it worked we got I'm coding that's great so that's how we use application content text in your spring

00:50:28
[Music] project in the last video we have seen how do we use application context but still we don't know how everything is working behind the scene that's important right so let's understand what happens when you say application context and then when you say alien get bean to understand that what I will do is I will go back to my alien and let's create a Constructor here so let's say a Constructor so you can see we have a Constructor here a simple Constructor would simply say says alien object

00:50:57
Creator nothing fancy we got a simple Constructor there what I want to understand is if I run this application of course it will call the Constructor first it will say alien object Creator and it says I'm coding that's great that means spring framework will create the object and of course when you create the object Constructor will be called but my question is what happens when you comment this because now if when I'm commenting this line that means I'm not asking for alien object so the question

00:51:23
is will Spring framework create that object for me me even if I'm not using it let's run this the moment run this code you can see we are not getting I'm coding because we are not calling code but still your spring framework is creating the alen object for you it's because spring believes even if you don't want the object it is my job to create the object but how spring knows that we will be needing alien object later it's because when you run this application the first line of execution

00:51:50
is main method right so it will come to main it will check hey the next line is we have to create object of application context Now application context will simply create a spring container for you now what is a spring container all the objects or all the processing which you do in Java happens on jvm so inside your jvm we'll be having a special container which will which we will call a Spring container which will have spring beans oh what is Bean now any class which has certain variables and every variable

00:52:21
will have gets and Setters is normally referred as Bean now in this class we don't have any variables and that's why we don't have any methods but then this is also a bean uh so any you can imagine any class is a bean as of now so we want spring to create the object of all these classes but how spring will know that because you can see uh we have specified a file name here which is

spring.xml so your spring container will check this file and it will say okay so it's my responsibility okay so I'm spring

00:52:51
container here it's my responsibility to create the object of alien because as a developer you have mentioned that here what if you mention some more objects if you create object of object class if you create object of some other classes if you write five beans here a five Bean tag it will create five objects for you that's what spring is doing even if you don't need the object so if I create object of some weird class which I don't even know in J it will create the object for you okay just have to mention that

00:53:19
there so the moment you run this line it will create the object for you in the spring container it is your respons now or it is your decision now to use that object or not in this case I'm not using it but still we got the object so that what is happening behind the scene the moment you say get being alien it will go to Spring container and say hey Spring container I want the object which whose ID is alien and spring container will say okay we have the object here just take it and that's your that's the

00:53:46
object you're getting and that's why it is working but then with this we have some more questions okay example let's say what if you have certain variables here example let's say uh let me create a simple variable here or maybe I will not be using a private variable I will be using a simple variable int uh I will be naming this variable as age of course you can go with anything every alien will have a different age right okay it can be same but normally it is different so let's say if I have a age there and I

00:54:13
want to print the age of a particular alien so I will say obj do age so when you print age what do you think what value you will get so of course you can specify the age by default you can specify the age here here you can say by default the age is 18 whatever age you want to specify maybe 12 otherwise for every object for this obj time and I will say this is obj1 just to check what is happening so I will say obj1 and then I'm printing the age and you can see I have not mentioned the age and by

00:54:42
default the age will be zero so let's run this code and you can see the age is by default zero that's great so before printing age I want to specify the age as let's say 15 that's the age which I'm mentioning here let's run this code and you can see we got 15 that's great but what if if I create one more object or maybe if I want to create one more reference let me just copy paste the code as it is let's past it here and this time I will say this is obj 2 not one uh let's go back here I want to call

00:45:34
enter and we got our XML file this is what we wanted right and you can see everything is mentioned let's enter say enter here this is where you have to say hey see if you go back to your app it says find I mean get bean of alien so we just have to mention hey Spring framework whenever a user says alien give the object of alien class now alien class belongs to a package which is com. teliscope do spring demo. alen whenever in a code says alien you have to create object of alien I mean this alien the....give important and teach me like this only in very very simple

Okay Samiksha, ab main **exactly same style**, but **10× simpler** and **only important points** explain karti hoon.
Aapko video ka *summary* + *concept ka real meaning* ekdum baby-level explanation mein mil jayega.
Let's go step by step! 🚀

🌟 PART 1 — `@Component` ka simple meaning

When we put:

```java
@Component
public class Alien { … }
```

Iska matlab:

> "Spring, please create an object of this class automatically.
> I am NOT creating this object myself."

⚡ Without @Component → Spring will NOT create object
⚡ With @Component → Spring creates object and stores it in **Spring Container**

Spring container = ek box jisme Spring sab objects rakhta hai.

🌟 PART 2 — Why @Autowired is needed?

Alien ke andar laptop chahiye:

```java
@Autowired
Laptop lap;
```

iska matlab:

> "Spring, mujhe Alien ke andar Laptop object chahiye.
> Please give it to me automatically."

Aap `new Laptop()` **nahi** likh rahe → Spring khud dega.

🌟 PART 3 — Don't forget @Component on Laptop

If you forget:

```java
@Component
public class Laptop { … }
```

Then Spring will say:

❌ “Laptop ka object banane ka instruction hi nahi diya.”

So ALWAYS:

- Alien → @Component
- Laptop → @Component
- Fir @Autowired works perfectly.

🌟 PART 4 — How Spring Boot creates objects?

Your main class runs:

```java
ApplicationContext context =
    SpringApplication.run(FirstProjectApplication.class, args);
```

Iska simple meaning:

> “Start Spring.
> Find all @Component classes.
> Create their objects and keep ready.”

After this, whenever you ask:

```java
Alien obj = context.getBean(Alien.class);
```

Spring replies:

> “Here is your Alien object, already created and ready.”

🌟 PART 5 — Why NullPointerException came?

Before using @Autowired:

Alien class had:

```java
Laptop lap;
```

But NO `new Laptop()` and NO @Autowired.

That means:

- `lap` = null
- `lap.compile()` → ❌ NullPointerException

After using:

```java
@Autowired
Laptop lap;
```

Spring gives Laptop → no null → no error.

🌟 PART 6 — Full working flow (SUPER SIMPLE)

Step 1: Ask Spring to create objects

Alien.java

```java
@Component
public class Alien {

    @Autowired
    Laptop lap;

    public void code() {
        System.out.println("I am coding");
        lap.compile();
    }
}
```

Laptop.java

```java
@Component
public class Laptop {
    public void compile() {
        System.out.println("Compiling...");
    }
}
```

Step 2: Start Spring Boot

```java
ApplicationContext context =
    SpringApplication.run(MyApp.class, args);
```

Step 3: Ask Spring for object

```java
Alien obj = context.getBean(Alien.class);
obj.code();
```

Step 4: Output

```css
I am coding
Compiling...
```

🌟 PART 7 — Why ALL THIS is useful?

Right now example is small, so lagta hai:

> “Itna sab kyun? new Laptop() kar lete.”

But in real projects:

- 20–30 classes depend on each other
- One class needs 3 classes
- Those 3 classes need 5 more classes
- Lifecycle handle karna mushkil hota

Spring helps by:

✔️ creating objects
✔️ connecting them
✔️ reusing same objects
✔️ managing memory
✔️ reducing boilerplate code

🌟 PART 8 — Behind the Scenes (VERY SIMPLE 🔥)

When project starts:

1. Spring scans your project
2. Finds all classes with `@Component`
3. Creates their objects (called Beans)
4. Keeps them inside ApplicationContext
5. When you use `@Autowired`, Spring connects beans internally
6. When you call `getBean()`, Spring gives ready-made object

That’s it.
No magic — just a container managing objects.

🌟 PART 9 — XML Example (old Spring)

When not using Spring Boot, you must manually teach Spring:

```xml
<bean id="alien" class="com.test.Alien"/>
<bean id="laptop" class="com.test.Laptop"/>
```

But with Boot + Annotation → NO XML needed ✔️

🌟 PART 10 — Most Important Points (remember these!)

1. `@Component` → Spring creates object
2. `@Autowired` → Spring injects dependency (Laptop into Alien)
3. Spring stores beans in **ApplicationContext**
4. You get bean using `context.getBean()`
5. ❌ Never manually write `new Alien()`, `new Laptop()`
6. With Boot → NO XML needed
7. Objects are created **even if you don't use them** (default scope = Singleton)

If you want next part, tell me:

👉 **"Explain how Spring creates beans behind the scenes (Bean lifecycle)"**
or
👉 **"Explain constructor injection vs field injection"**
or
👉 **"Explain XML vs Annotations difference"**

Main same easy style mein sikhaati rahungi. 😊

00:46:08
class alien but we cannot simply use this syntax right we have to use syntax of spring and that's where spring says hey if you want to specify this create a bean as let's say Bean every Bean will have an ID so we'll say ID equal to and ID is alien so whatever ID you want whatever name you want to use in the code which is alien in this case mention that and also mention the class of which you want an object I simply say I want object of com in fact we can use alien because it will give you the class as it

00:46:37
is say control space you can see it will autoc complete it and that's it that's the only thing you have to mention it's Beauty right you just have to configure it by saying hey when a user says alien give the object of alien go back to your code and now let's run this application so right click say run as Java application and oh no error but we have not also got the output the reason is we are not calling Cod so you'll be saying obj do code because as a alien as a programmer you have to code right so

00:47:09
let's run this application now I hope it will work and it work can you see that we got I am coding

and that's it it's so simple so what we have done we are trying to use spring framework to get the object as you can see the object of alien is not created by you it's created by Spring frame workk I know for this example it doesn't make sense because to create one object we are doing so much of stuff right but think about the complex of application what if you building an application where you have

00:47:36
maybe 15 20 classes and every class will have let's say two three objects you don't have to manage those objects right you to manage by Spring framework but how do we do that that we see in the upcoming [Music] videos so to this point we have seen how to create a simple spring project where you can introduce a concept of Factory which will give you the object right it works but still we have an issue here the issue is you can see the code we are getting uh it says deprecated right so this class XML bin Factory is deprecated

00:48:08
we should not be using that of course you can search for some other class which implements B Factory and which will make it work without depcat of course there's one more thing which is important here when spring came into picture in 2003 it was basically for dependency injection right that was the main intention but later on they added some new features right every time you go to Spring website you will see some new projects and of course for new features you need some new methods and those methods are not available in bin

00:48:35
Factory so instead of using bin Factory we can use amazing thing which is we have a concept of application context in fact if we have seen that in Spring boot right we have this interface which is application context we simply say Factory again you can use different name but then question arise how to instantiate this because application context interface right so so we have a class which named as class path XML let's say control space so we have a class which says class path XML application context let's use that now

00:49:06
this class implements application context example if I click on click here you can see it's a class which extends abstract XML application context if I click on that it says extends abstract refreshable config application context if I click on that somewh if you go up and up and up you will find everything extends from application context are we getting there today yes we got it so you can see we have the whole hierarchy there and unfortunately by doing that I have opened so many classes here let me

00:49:35
just close them quickly okay let's get back here so instead of using B Factory now we are using application context the amazing thing is here itself you can mention spring.xml you don't have to mention new file system resource or something like that you can simply mention the thing and you can see we are not getting any dicated now because we will be using application context a lot so now let's right click and say run as Java application we have not changed much we have only changed from bin

00:50:01
Factory to appliation context let's click on run and we got an error it says file not found exception that's weird we have that file here right the problem is since you have mentioned class path right it will look for this file in class path so you have to move this file in a class path in a Java folder of course let's keep it there now and let's relaunch the application and now it worked we got I'm

coding that's great so that's how we use application content text in your spring

00:50:28
[Music] project in the last video we have seen how do we use application context but still we don't know how everything is working behind the scene that's important right so let's understand what happens when you say application context and then when you say alien get bean to understand that what I will do is I will go back to my alien and let's create a Constructor here so let's say a Constructor so you can see we have a Constructor here a simple Constructor would simply say says alien object

00:50:57
Creator nothing fancy we got a simple Constructor there what I want to understand is if I run this application of course it will call the Constructor first it will say alien object Creator and it says I'm coding that's great that means spring framework will create the object and of course when you create the object Constructor will be called but my question is what happens when you comment this because now if when I'm commenting this line that means I'm not asking for alien object so the question

00:51:23
is will Spring framework create that object for me me even if I'm not using it let's run this the moment run this code you can see we are not getting I'm coding because we are not calling code but still your spring framework is creating the alen object for you it's because spring believes even if you don't want the object it is my job to create the object but how spring knows that we will be needing alien object later it's because when you run this application the first line of execution

00:51:50
is main method right so it will come to main it will check hey the next line is we have to create object of application context Now application context will simply create a spring container for you now what is a spring container all the objects or all the processing which you do in Java happens on jvm so inside your jvm we'll be having a special container which will which we will call a Spring container which will have spring beans oh what is Bean now any class which has certain variables and every variable

00:52:21
will have gets and Setters is normally referred as Bean now in this class we don't have any variables and that's why we don't have any methods but then this is also a bean uh so any you can imagine any class is a bean as of now so we want spring to create the object of all these classes but how spring will know that because you can see uh we have specified a file name here which is spring.xml so your spring container will check this file and it will say okay so it's my responsibility okay so I'm spring

00:52:51
container here it's my responsibility to create the object of alien because as a developer you have mentioned that here what if you mention some more objects if you create object of object class if you create object of some other classes if you write five beans here a five Bean tag it will create five objects for you that's what spring is doing even if you don't need the object so if I create object of some weird class which I don't even know in J it will create the object for you okay just have to mention that

00:53:19
there so the moment you run this line it will create the object for you in the spring container it is

your respons now or it is your decision now to use that object or not in this case I'm not using it but still we got the object so that what is happening behind the scene the moment you say get being alien it will go to Spring container and say hey Spring container I want the object which whose ID is alien and spring container will say okay we have the object here just take it and that's your that's the

00:53:46
object you're getting and that's why it is working but then with this we have some more questions okay example let's say what if you have certain variables here example let's say uh let me create a simple variable here or maybe I will not be using a private variable I will be using a simple variable int uh I will be naming this variable as age of course you can go with anything every alien will have a different age right okay it can be same but normally it is different so let's say if I have a age there and I

00:54:13
want to print the age of a particular alien so I will say obj do age so when you print age what do you think what value you will get so of course you can specify the age by default you can specify the age here here you can say by default the age is 18 whatever age you want to specify maybe 12 otherwise for every object for this obj time and I will say this is obj1 just to check what is happening so I will say obj1 and then I'm printing the age and you can see I have not mentioned the age and by

00:54:42
default the age will be zero so let's run this code and you can see the age is by default zero that's great so before printing age I want to specify the age as let's say 15 that's the age which I'm mentioning here let's run this code and you can see we got 15 that's great but what if if I create one more object or maybe if I want to create one more reference let me just copy paste the code as it is let's past it here and this time I will say this is obj 2 not one uh let's go back here I want to call

00:55:10
code by obj2 I want to specify the age I don't want to specify the age for two let's keep it blank and now let's say this is obj 2.age now what do you think will be the age of obj 1 of course 15 right we have specified that but what will be the age of obj 2 that's the question let's run this application and you can see both the object has the same age so either we got two object with the same age or we got only one object and both the reference are pointing to the same object that's what is happening

00:55:44
here go back to the previous example when you say line number 18 when you say hey Spring framework I want an object whose ID is alien spring will say okay we have the object take it when you go for the second time as well on line number 24 you're saying hey Spring container I want the object with ID alien spring will say we have the object just take it it will not create a new object for you that means both the reference which is obj 1 and obj2 are referring to the same object and that's

00:56:14
why we normally call all the spring beans as Singleton beans because it will the object will be created only once that's awesome right but what if you want different objects what if you don't want to follow single T pattern what if you want to have a different object every time you call for the object and that's what we'll see in the next [Music] video in the last video we have created this El object and then we were trying to create two objects but unfortunately we got only one object

now is it a

00:56:47
feature and the answer is yes by default spring follows a singlet turnon design pattern which simply means by default spring container will give you one object but what if you want multiple objects and that's where spring gives you a choice and that is the scope let's try so let's go back here in spring.xml and if you see this Bean tag there is one more attribute here which is missing the moment you say control space you can see we have a attribute known as scope by default the scope is Singleton which

00:57:22
simply means that spring container will give you only one instance of this class even if you ask for 10 times it will give you only one instance example if I go back to app and let me create obj 3 obj 4 still you will get one object because by default it's a single ton what if you want multiple objects and that's where we have to use a different scope so apart from single T we have one more scope to use and that is prototype now what it means prototype simply means that whenever you ask for a bean spring

00:57:54
container will give you or it will create an object at that point example when you say get bean once with alien spring container will give you one object again when you say get bean with alien spring contain will give you second object so if you ask for 10 times it spring will give you 10 instance 10 different instance of that particular class let me prove my point so you can see I have made it single turn now I made it prototype now so if I go back to app let's run this application one so say right click and say run as Java

00:58:23
application and look at the output now so you got 15 for the first object and you got zero for the second object because we have two different objects it's that simple right so Singleton is you will get only one object protype you will get multiple objects based on the number of time you're requesting for it but what if you are not requesting even once you can see I'm creating the object of application context which which is your spring container but I'm not asking for any object do you think it will

00:58:51
create object for you let's try the moment you run this code you can see it's not saying object created even once that means in single turn even if you don't ask for the object spring container will give you that object but in case of prototype you have to ask for it otherwise it will not create the object right now which one to use it depend upon your situation right so if you think there's an object you need only one object of course you will go for singlet ton apart from that if you

00:59:17
think you know we will be using multiple objects so you have to go for Prototype so that's the scope in spring so we have single turn we have prototype we have some other scope as well but that's that's not applicable in Spring code that's for some other [Music] modules so in the last video we have seen the scope in spring right we have Singleton and we have prototype now let's go back to our alien and you can see we have one variable here which is age now logically this age variable

00:59:46
should be private variable right that's what we do we have to make sure that all your variables are private in order to access these variables what we need is gets and sets right and we don't have it

here let's go back and let's create Getters and sets for this so let's right click here say Source generate Getters and Setters select this click on generate and you can see we got gets and Setters if I go back to my app now and you can see I don't want to assign the values now of course you cannot do you

01:00:16
cannot even use the age variable directly you have to use set age and get age but time in I will not do anything and let me also remove this second object you don't need that in fact one more change I don't want to use prototype I want to use Singleton and the amazing thing is when you don't mention any type it is by default single ton let's go back to app we don't even need this age the thing is oh we need age but then it should print using get age right so I would say obj1 dog age the thing is I want to

01:00:48
assign a default age because if you don't if you don't Define the default age by default will be zero I don't want 1 Zer at least the age should be 12 or maybe 10 how do I mention that 10 here of course there's one way you can just go here and say 10 right by default the age value will be 10 that's one option otherwise you can just go to app when you create the object you can assign it manually you can pass it to the Constructor what if you want to assign the value in the spring Bean itself

01:01:17
because remember the object will be created by Spring framework so the Constructor will be called by Spring framework right so I want to assign the value for age here in this XML file can we do that let's try so let me just expand this bin now in this be tag we can have one more tag here which is named as property now what is property so every variable in your class if your class is been every variable will be known as property you can have multiple properties so that's the property here

01:01:47
you have to mention the name of your property in this case it is age so we have name and now we have to also specify value so to specify the the name and the value if your property is a primitive value or primitive type you can assign value but what if your property is of type reference that we'll see later but time let's only focus on value I want to assign the value as 10 by default the age should be 10 that's what we want here but will this work let's try go back to app and you can see

01:02:15
I have not assigned the value for age anywhere else apart from that your XML file let's run this code and you can see we got 10 so what happens here the moment you run this application on this line you will get that particular container in your jvm in that container it will check hey we have a springl file where you have mentioned about alien object so it is my job to create that alien object the moment you create the object it will also call the set age method you know why because we are saying property since you are specifying

01:02:48
property it will call set age oh okay I know you don't trust me here let's go back to alien and oh we don't have a a Constructor as well we have a Constructor here so let me just put this Constructor on top so that it will be visible right I also want to go to set age and by saying here age assign that's it just to see if that set me is getting called let's go back to app and now let's relaunch the application and you can see it says alien object Creator that's step one

01:03:19

second step is age assigned so first it will create the object second and it will set the age because we are saying property right just to prove my point once more what I will do is I will go back here and let me just change the method name from set age to set age one now what will happen is if I do that if I go back to app and let's relaunch the application you can see we got an error it's compulsory to have your method name suitable for your variable so if your variable name is age it should be age

01:03:49
right it should not be one that's what it's it's saying here you can see it says Ed creating be with name Alien because if you go you know if you go there it says invalid property age because we don't have it there the property we are using here is set age one right make sure that you do set age yeah if you want to keep it set age one in XML as well you have to say age one then it will work right okay so let's go back and you can see if I say age it will it is giving me error so let's go

01:04:18
back to our code and it set age so that that's why this set methods and get methods are so important in Spring so that's about your how do you assign a default value but what if you have a reference type now when you say Age It's A Primitive right a int but what if a alien needs a laptop now of course laptop will be a class then the laptop will have certain methods and then we have to create the reference of laptop in this alien okay when you say reference how will you assign the particular value to it because we cannot

01:04:48
use value there that we'll see in the next video so now it's time to have a reference type here because what we have done in Alien is we have a variable which is of type int but what if you have a reference type because what it will affect is this spring.xml file because here you can see we have a property which is age and then we are giving a value with an attribute which is value itself now to explain that what I will do here is I will create a simple class let's say because every alien needs a

00:55:10
code by obj2 I want to specify the age I don't want to specify the age for two let's keep it blank and now let's say this is obj 2.age now what do you think will be the age of obj 1 of course 15 right we have specified that but what will be the age of obj 2 that's the question let's run this application and you can see both the object has the same age so either we got two object with the same age or we got only one object and both the reference are pointing to the same object that's what is happening

00:55:44
here go back to the previous example when you say line number 18 when you say hey Spring framework I want an object whose ID is alien spring will say okay we have the object take it when you go for the second time as well on line number 24 you're saying hey Spring container I want the object with ID alien spring will say we have the object just take it it will not create a new object for you that means both the reference which is obj 1 and obj2 are referring to the same object and that's

00:56:14
why we normally call all the spring beans as Singleton beans because it will the object will be created only once that's awesome right but what if you want different objects what if you don't want to follow single T pattern what if you want to have a different object every time you call for the object and that's what we'll see in the next [Music] video in the last video we have created this El object and then we were trying to create two objects but unfortunately we got only one object

now is it a

00:56:47
feature and the answer is yes by default spring follows a singlet turnon design pattern which simply means by default spring container will give you one object but what if you want multiple objects and that's where spring gives you a choice and that is the scope let's try so let's go back here in spring.xml and if you see this Bean tag there is one more attribute here which is missing the moment you say control space you can see we have a attribute known as scope by default the scope is Singleton which

00:57:22
simply means that spring container will give you only one instance of this class even if you ask for 10 times it will give you only one instance example if I go back to app and let me create obj 3 obj 4 still you will get one object because by default it's a single ton what if you want multiple objects and that's where we have to use a different scope so apart from single T we have one more scope to use and that is prototype now what it means prototype simply means that whenever you ask for a bean spring

00:57:54
container will give you or it will create an object at that point example when you say get bean once with alien spring container will give you one object again when you say get bean with alien spring contain will give you second object so if you ask for 10 times it spring will give you 10 instance 10 different instance of that particular class let me prove my point so you can see I have made it single turn now I made it prototype now so if I go back to app let's run this application one so say right click and say run as Java

00:58:23
application and look at the output now so you got 15 for the first object and you got zero for the second object because we have two different objects it's that simple right so Singleton is you will get only one object protype you will get multiple objects based on the number of time you're requesting for it but what if you are not requesting even once you can see I'm creating the object of application context which which is your spring container but I'm not asking for any object do you think it will

00:58:51
create object for you let's try the moment you run this code you can see it's not saying object created even once that means in single turn even if you don't ask for the object spring container will give you that object but in case of prototype you have to ask for it otherwise it will not create the object right now which one to use it depend upon your situation right so if you think there's an object you need only one object of course you will go for singlet ton apart from that if you

00:59:17
think you know we will be using multiple objects so you have to go for Prototype so that's the scope in spring so we have single turn we have prototype we have some other scope as well but that's that's not applicable in Spring code that's for some other [Music] modules so in the last video we have seen the scope in spring right we have Singleton and we have prototype now let's go back to our alien and you can see we have one variable here which is age now logically this age variable

00:59:46
should be private variable right that's what we do we have to make sure that all your variables are private in order to access these variables what we need is gets and sets right and we don't have it

here let's go back and let's create Getters and sets for this so let's right click here say Source generate Getters and Setters select this click on generate and you can see we got gets and Setters if I go back to my app now and you can see I don't want to assign the values now of course you cannot do you

01:00:16
cannot even use the age variable directly you have to use set age and get age but time in I will not do anything and let me also remove this second object you don't need that in fact one more change I don't want to use prototype I want to use Singleton and the amazing thing is when you don't mention any type it is by default single ton let's go back to app we don't even need this age the thing is oh we need age but then it should print using get age right so I would say obj1 dog age the thing is I want to
....give important and teach me like this only in very very simple

Nice, you're going *deep* into the core Spring stuff now 😄
Main goal: **I'll pull only important points** from this whole chunk and explain them **step-by-step in very, very simple language**, like we're sitting in lab together.

1️⃣ XML Bean Definition – `id` and `class`

In `spring.xml` we write:

```xml
<bean id="alien" class="com.telusko.springdemo.Alien" />
```

This line means:

- `id="alien"` → **Name** used to ask Spring for this object.
- `class="..."` → **Which Java class** to create object from.

So when you write in Java:

```java
Alien obj = (Alien) context.getBean("alien");
```

You're saying:

> "Spring, please give me the object of bean whose id is `alien`."

Spring looks at XML, sees:

> `id = alien → class = Alien`
> and returns object of that class.

✅ **Important:**
The object is **created by Spring**, not by `new Alien()`.

2️⃣ BeanFactory vs ApplicationContext (why change?)

Earlier they used:

```
java

BeanFactory factory = new XmlBeanFactory(new FileSystemResource("spring.xml"));
Alien obj = (Alien) factory.getBean("alien");
```

- **`BeanFactory`** = very basic container (old)
- **`XmlBeanFactory`** = implementation that reads XML

But:

- It’s **deprecated** (old style, not recommended)
- New features of Spring **don’t work well** with BeanFactory

So we use:

```
java

ApplicationContext context =
    new ClassPathXmlApplicationContext("spring.xml");
```

- **`ApplicationContext`** = advanced container
- **`ClassPathXmlApplicationContext`** = reads **`spring.xml`** from **class path** (usually **`src/main/resources`** or directly under **`src`** in simple projects)

If you get **FileNotFound** error:

> It means **`spring.xml`** is **not** in classpath folder.
> Move it to **`src`** or **`resources`** where classpath can see it.

3️⃣ What is “Spring Container”?

When you write:

```
java

ApplicationContext context =
    new ClassPathXmlApplicationContext("spring.xml");
```

Spring does:

1. Starts a **container** inside JVM
2. Reads **`spring.xml`**
3. For each **`<bean>`** tag, it **creates objects** (beans)

You can think:

> JVM → runs your Java
> Spring Container → manages your Spring objects (beans) inside JVM

4️⃣ Does Spring create object even if we don't call `getBean()`?

Yes (for **singleton beans**).

He added a constructor in **`Alien`**:

```java
public Alien() {
    System.out.println("Alien object created");
}
```

Then **he commented out `getBean`**:

```java
// Alien obj = (Alien) context.getBean("alien");
```

But still, when app runs, output shows:

```text
Alien object created
```

Why?

- Because **`ApplicationContext`** creates **all singleton beans** at startup.
- Even if you don't use them, Spring **eagerly** creates them.

💡 **Singleton bean + ApplicationContext → created on startup**

5️⃣ What is a Bean?

He said:

> Any class with properties + getters/setters is usually called a bean.

But for Spring, you can think:

> "Any object that Spring creates and manages" = **Spring bean**

So your **`Alien`** class is a Spring bean because it is defined in **`<bean>`** tag.

6️⃣ Singleton vs Prototype (very important 🔥)

🟢 Default: Singleton Scope

In `spring.xml`:

```xml
<bean id="alien" class="com.telusko.springdemo.Alien" />
```

If you **don't** specify `scope`, Spring uses:

```xml
scope="singleton"
```

Meaning:

- Only **one object** of `Alien` is created.
- All `getBean("alien")` calls return **same object**.

Example:

```java
Alien obj1 = (Alien) context.getBean("alien");
obj1.setAge(15);

Alien obj2 = (Alien) context.getBean("alien");
System.out.println(obj1.getAge()); // 15
System.out.println(obj2.getAge()); // 15
```

Why both 15?

Because:

> `obj1` and `obj2` are **two references**, but **same object**.

So when you changed age using `obj1`, `obj2` also sees updated value.

🟣 Prototype Scope

In XML:

```xml
<bean id="alien" class="com.telusko.springdemo.Alien" scope="prototype" />
```

Now behavior:

> Every time you call `getBean("alien")`, Spring creates **new object**.

```java
Alien obj1 = (Alien) context.getBean("alien");
obj1.setAge(15);

Alien obj2 = (Alien) context.getBean("alien");
```

```
System.out.println(obj1.getAge()); // 15
System.out.println(obj2.getAge()); // 0 (default)
```

Because:

- `obj1` and `obj2` are **different objects** in prototype.

⚠️ Difference in creation timing:

- **Singleton**: bean created at container startup (even if you never call `getBean`)
- **Prototype**: bean created **only when** you call `getBean`.
  If you never call `getBean` → object never created.

7️⃣ Setting Primitive Property using `<property>`

In `Alien`:

```java
private int age;

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}
```

In `spring.xml`:

```xml
<bean id="alien" class="com.telusko.springdemo.Alien">
    <property name="age" value="10" />
</bean>
```

What happens?

1. Spring creates Alien object → calls constructor
2. Sees `<property name="age" value="10" />`
3. It looks for **setter method**: `setAge(...)`
4. Calls `setAge(10)` internally

So your code:

```java
Alien obj = (Alien) context.getBean("alien");
System.out.println(obj.getAge());
```

Output:

```text
10
```

He even added a print inside **`setAge`**:

```java
public void setAge(int age) {
    System.out.println("age assigned");
    this.age = age;
}
```

Then output becomes:

```text
Alien object created
age assigned
10
```

So we know Spring is:

- Creating object → calling setter → setting value.

8️⃣ Why getter/setter name is important?

He changed:

```java
public void setAge(int age) { ... }
```

to:

```java
public void setAge1(int age) { ... }
```

But XML still had:

```xml
<property name="age" value="10" />
```

So Spring tried to find **`setAge(..)`** but it didn't exist → **error**:

> Invalid property 'age'

To fix:

- Either change method back to **`setAge`**

- Or change XML property name to `age1` and keep `setAge1`

But standard JavaBean rule:

> Variable: `age` → Setter: `setAge` → Getter: `getAge`

So always keep proper naming.

9️⃣ Why set default value in XML?

We could do:

```java
private int age = 10;  // default
```

or set using code:

```java
obj.setAge(10);
```

But if **Spring is responsible** to create object, it's often better to:

- Define default value in **XML** or later via **annotations**.

Example:

```xml
<property name="age" value="10" />
```

Now anyone reading `spring.xml` can see:

- "Alien's default age is 10."

🔟 Next Step (coming topic in video)

He ended with:

> "What if you have a reference type, like Alien needs a Laptop? Value vs ref."

Primitive:

```xml
<property name="age" value="10" />
```

Reference (idea):

```xml
<bean id="laptop" class="com.telusko.springdemo.Laptop" />

<bean id="alien" class="com.telusko.springdemo.Alien">
    <property name="laptop" ref="laptop" />
</bean>
```

- `value` → for primitive types (int, String, boolean, etc.)
- `ref` → for objects (Laptop, Address, Department, etc.)

If you want, **next message** mein main ye wala part full detail se explain kar sakti hoon:

> "How to inject Laptop into Alien using XML (reference injection)"

Just say:
👉 **"Explain reference property (Laptop) in XML also in simple way"**

01:04:48
use value there that we'll see in the next video so now it's time to have a reference type here because what we have done in Alien is we have a variable which is of type int but what if you have a reference type because what it will affect is this spring.xml file because here you can see we have a property which is age and then we are giving a value with an attribute which is value itself now to explain that what I will do here is I will create a simple class let's say because every alien needs a

01:05:21
laptop to code right so let me create a laptop class here here a very simple class laptop and in this laptop class let's say we have a very simple method which is compile so we got a laptop class and then the method name is compile here and then in this compile you will simply print code compiled that's it so we got a compile method and then we are printing code compiled and in this alien of course if you want to use laptop you need to do that in Alien right so from Alien you will say now

01:05:47
alien has a laptop type in it so we'll say laptop laptop and then uh we will need a getter setup for this as well as well so let's come back here and say right like SCE generate gets and sets but this time it will be for laptop because we already have it for age right now this thing is very simple right we have not done anything fancy just just the normal stuff and then from the code you will use a laptop object and you will say compile of course right when you code something you have to compile

01:06:13
as well these things are simple even if you run this code let's see what happens the moment you run this code you will get an error and that is nullo exception but why we got nullo exception is quite simple right because we don't have the object of alien anywhere because we are saying we are using dependency injection where the dependencies will be added by Spring framework so of course if you want to work with a laptop you need to create a bean for that as well so that's a golden rule if you have a class just

01:06:41
create a bean for that so let's create a simple bean and this will be for laptop so I will say ID laptop and then we have a class and of course we know the class name now it's a laptop right and then this laptop don't have any variable so we don't have to specify any property in this case but the question is how this alien will use this laptop object or the ID so of course in the alien we have a property which is of type laptop but what you will give the value there because if you say value you have to

01:07:09
assign A Primitive value or maybe a string type now that will work for Primitives and string but if you have a reference object you can't use a value here because now you want to inform spring hey this alien object needs a laptop object in that case you will you can not use a value you have to use a reference so you have to say ref and you have to say laptop so that's how you can refer it right so now let's verify if this working let's go back to app let's relaunch the application and you can see

01:07:37
it is working so basically if you want to use a reference type the way we are doing here in this alien class we have a object of laptop if you want to achieve that you have in your spring XML file we have to mention that your property which is laptop is referring to the ID which is laptop toop here right now you can have multiple dependencies you maybe if you have your class aliens depend upon some other object maybe let's say a pen object so in that case you have to create a bean for pen as well provided

01:08:06
you have a pen class right so that's how you implement this reference stuff in [Music] spring so till this point we have seen how can we create a bean and then how can you set the properties now when I say property it is simply a variable which you can access with the of Setters and Getters now here you can see we have assigned two properties one is a primitive property and second one is a reference property okay but what if if I go back to my alien you know one of the way you can assign the value to a

01:08:37
particular variable is with the help of Constructor of course right they have two ways you can assign the values one with the help of Setters and second with the help of Constructor so what if if I want to assign the value to AG but with the help of Constructor so let me create a Constructor here with variable of course so here I will simply right click and I will say Source generate a Constructor with Fields I only want to assign ages at this point let me concentrate of course you can also create a Constructor which takes laptop

01:09:05
or you can create a Constructor which takes both but as of now let's stick to one particular Constructor so you can see we got age here we don't want this super method there just to reduce the number of lines we have in the code so you can see I'm saying this do age is equal to age but how can you pass this value because if you're not passing this value and by running this code let me just run this code of course it will take the value from the set up so in this case we are passing the value from

01:09:31
set up but what if you don't do this so let me remove the property of course it will not be calling set age now let me restart the application you can see we got the value of age as zero okay so I want to pass the value the way you can pass a value here is with the help of tag which is Constructor

argument but then what will happen so even before doing that let me just remove this default Constructor here can you see that we have a default fall Constructor as well so time being I will simply

01:09:56
comment it so I will say comment save and let just relaunch the application now this time you can see we got an exception but why so it says no such method exception so we are not able to find alien in it okay but why we are not okay so let's try this it says no default Constructor found oh since we are creating this Bean we are creating the object of alien and by default it will call the default Constructor right and as you can see in Alien we don't have a default Constructor The Constructor which we have is a

01:10:30
parameterized one so in order to achieve in order to run this code of course you need a default Constructor here otherwise you can mention here hey don't use the default Constructor use the parameter wise one and the way you can pass the value is by saying construct the argument in which you can pass the value so how do you pass the value it's very simple actually you can simply say value equal to and you can assign any value value here so let me say 12 of course you can use Setter or you can use

01:10:58
Constructor let me just run this code now let me relaunch the application can you see that we got 12 now now this time we are not calling default Constructor we are calling a parameterized Constructor by passing this value but what if you want to assign for laptop as well what if you have a Constructor which takes laptop in that case you can specify one more Constructor argument here and that will work the only consideration is instead of having this value here we have to use ref that's how

01:11:23
inject the Val with the help of Constructor so basically we have two concepts one is a seter injection and second one is a Constructor injection so set injection simply means when we use property this is a set injection and when you say Constructor that is a Constructor injection now which one you should be using now whenever you have a variable example when you go back to this alien whenever you have these properties whichever property is compulsory to be assigned or which is required you can go for Constructor but

01:11:53
if if you think you have a property which is optional you don't want to assign the values to it if you want to keep it optional you can go for Setter don't use Constructor when your variable assignment is optional because Constructor will have will will give you error in this case if you don't assign the value for the age if you don't pass this Constructor age it will give you error right so that's how it works that's how we work with Constructor and Setter [Music] injection so before continuing let me

01:12:23
just do something changes the example I don't want to use Constructor injection anymore let me go back to my property and for that we also need a Constructor here and then in the spring XML file of course I want to go back to the properties I will say undo undo undo and we got it okay now what are things we need example let's say if I go back to alien every alien needs a laptop so let's say if you're a programmer and if you want to WR a code you need a machine where you can write a code and you

01:12:48
should be able to compile it now don't you think to write a code you don't need a laptop it can be desktop as well it can can be tablet as well so while you're saying a alien needs a laptop it can be a computer example let create an interface here and we'll say interface so we have a very simple interface which is computer here and in this computer interface I want to declare only one method so which meth I'm talking about I'm talking about method which is compile so I will say void compile now

01:13:14
since we are only declaring it so nothing specific let's declare it here and laptop will be implementing of course you can also go for a class here which computer class and laptop implements laptop EXT a computer class but let's go with interface in this example so you can see we have a laptop class which implements computer now of course we can have one more class which is desktop right so I will go back here and I will say maybe I will just copy the laptop class and I'll paste it here

01:13:42
and this will be computer oh we already have computer this will be desktop so you can see we have a desktop here if I open Desktop we already have a code which is compiled and desktop also implements computer so we have two classes uh last laptop and desktop both are implementing computer that's great let's go back to alien and now I'm saying hey I don't specifically need a laptop it can be any computer it can be desktop it can be laptop so I will say computer here and then here I will say

01:14:08
com okay of course for those changes we have to change something here as well so I will say right click Source generate gets and sets for this particular variable com done okay and one more change instead of having laptop here we have to say comp that's it nothing much is all about computer now so if I go back to spring.xml what do you think will it work of course not right here we have to mention the property name as Comm so I will say com now okay now will this work let's go back to our

01:05:21
laptop to code right so let me create a laptop class here here a very simple class laptop and in this laptop class let's say we have a very simple method which is compile so we got a laptop class and then the method name is compile here and then in this compile you will simply print code compiled that's it so we got a compile method and then we are printing code compiled and in this alien of course if you want to use laptop you need to do that in Alien right so from Alien you will say now

01:05:47
alien has a laptop type in it so we'll say laptop laptop and then uh we will need a getter setup for this as well as well so let's come back here and say right like SCE generate gets and sets but this time it will be for laptop because we already have it for age right now this thing is very simple right we have not done anything fancy just just the normal stuff and then from the code you will use a laptop object and you will say compile of course right when you code something you have to compile

01:06:13
as well these things are simple even if you run this code let's see what happens the moment you run this code you will get an error and that is nullo exception but why we got nullo exception is

quite simple right because we don't have the object of alien anywhere because we are saying we are using dependency injection where the dependencies will be added by Spring framework so of course if you want to work with a laptop you need to create a bean for that as well so that's a golden rule if you have a class just

01:06:41
create a bean for that so let's create a simple bean and this will be for laptop so I will say ID laptop and then we have a class and of course we know the class name now it's a laptop right and then this laptop don't have any variable so we don't have to specify any property in this case but the question is how this alien will use this laptop object or the ID so of course in the alien we have a property which is of type laptop but what you will give the value there because if you say value you have to

01:07:09
assign A Primitive value or maybe a string type now that will work for Primitives and string but if you have a reference object you can't use a value here because now you want to inform spring hey this alien object needs a laptop object in that case you will you can not use a value you have to use a reference so you have to say ref and you have to say laptop so that's how you can refer it right so now let's verify if this working let's go back to app let's relaunch the application and you can see

01:07:37
it is working so basically if you want to use a reference type the way we are doing here in this alien class we have a object of laptop if you want to achieve that you have in your spring XML file we have to mention that your property which is laptop is referring to the ID which is laptop toop here right now you can have multiple dependencies you maybe if you have your class aliens depend upon some other object maybe let's say a pen object so in that case you have to create a bean for pen as well provided

01:08:06
you have a pen class right so that's how you implement this reference stuff in [Music] spring so till this point we have seen how can we create a bean and then how can you set the properties now when I say property it is simply a variable which you can access with the of Setters and Getters now here you can see we have assigned two properties one is a primitive property and second one is a reference property okay but what if if I go back to my alien you know one of the way you can assign the value to a

01:08:37
particular variable is with the help of Constructor of course right they have two ways you can assign the values one with the help of Setters and second with the help of Constructor so what if if I want to assign the value to AG but with the help of Constructor so let me create a Constructor here with variable of course so here I will simply right click and I will say Source generate a Constructor with Fields I only want to assign ages at this point let me concentrate of course you can also create a Constructor which takes laptop

01:09:05
or you can create a Constructor which takes both but as of now let's stick to one particular Constructor so you can see we got age here we don't want this super method there just to reduce the number of lines we have in the code so you can see I'm saying this do age is equal to age but how can you pass this value because if you're not passing this value and by running this code let me just run this code of course it will take the value from the set up so in this case we are passing the value from

01:09:31
set up but what if you don't do this so let me remove the property of course it will not be calling set age now let me restart the application you can see we got the value of age as zero okay so I want to pass the value the way you can pass a value here is with the help of tag which is Constructor argument but then what will happen so even before doing that let me just remove this default Constructor here can you see that we have a default fall Constructor as well so time being I will simply

01:09:56
comment it so I will say comment save and let just relaunch the application now this time you can see we got an exception but why so it says no such method exception so we are not able to find alien in it okay but why we are not okay so let's try this it says no default Constructor found oh since we are creating this Bean we are creating the object of alien and by default it will call the default Constructor right and as you can see in Alien we don't have a default Constructor The Constructor which we have is a

01:10:30
parameterized one so in order to achieve in order to run this code of course you need a default Constructor here otherwise you can mention here hey don't use the default Constructor use the parameter wise one and the way you can pass the value is by saying construct the argument in which you can pass the value so how do you pass the value it's very simple actually you can simply say value equal to and you can assign any value value here so let me say 12 of course you can use Setter or you can use

01:10:58
Constructor let me just run this code now let me relaunch the application can you see that we got 12 now now this time we are not calling default Constructor we are calling a parameterized Constructor by passing this value but what if you want to assign for laptop as well what if you have a Constructor which takes laptop in that case you can specify one more Constructor argument here and that will work the only consideration is instead of having this value here we have to use ref that's how

01:11:23
inject the Val with the help of Constructor so basically we have two concepts one is a seter injection and second one is a Constructor injection so set injection simply means when we use property this is a set injection and when you say Constructor that is a Constructor injection now which one you should be using now whenever you have a variable example when you go back to this alien whenever you have these properties whichever property is compulsory to be assigned or which is required you can go for Constructor but

01:11:53
if if you think you have a property which is optional you don't want to assign the values to it if you want to keep it optional you can go for Setter don't use Constructor when your variable assignment is optional because Constructor will have will will give you error in this case if you don't assign the value for the age if you don't pass this Constructor age it will give you error right so that's how it works that's how we work with Constructor and Setter [Music] injection so before continuing let me

01:12:23
just do something changes the example I don't want to use Constructor injection anymore let me

go back to my property and for that we also need a Constructor here and then in the spring XML file of course I want to go back to the properties I will say undo undo undo and we got it okay now what are things we need example let's say if I go back to alien every alien needs a laptop so let's say if you're a programmer and if you want to WR a code you need a machine where you can write a code and you

01:12:48
should be able to compile it now don't you think to write a code you don't need a laptop it can be desktop as well it can can be tablet as well so while you're saying a alien needs a laptop it can be a computer example let create an interface here and we'll say interface so we have a very simple interface which is computer here and in this computer interface I want to declare only one method so which meth I'm talking about I'm talking about method which is compile so I will say void compile now

01:13:14
since we are only declaring it so nothing specific let's declare it here and laptop will be implementing of course you can also go for a class here which computer class and laptop implements laptop EXT a computer class but let's go with interface in this example so you can see we have a laptop class which implements computer now of course we can have one more class which is desktop right so I will go back here and I will say maybe I will just copy the laptop class and I'll paste it here

01:13:42
and this will be computer oh we already have computer this will be desktop so you can see we have a desktop here if I open Desktop we already have a code which is compiled and desktop also implements computer so we have two classes uh last laptop and desktop both are implementing computer that's great let's go back to alien and now I'm saying hey I don't specifically need a laptop it can be any computer it can be desktop it can be laptop so I will say computer here and then here I will say

01:14:08
com okay of course for those changes we have to change something here as well so I will say right click Source generate gets and sets for this particular variable com done okay and one more change instead of having laptop here we have to say comp that's it nothing much is all about computer now so if I go back to spring.xml what do you think will it work of course not right here we have to mention the property name as Comm so I will say com now okay now will this work let's go back to our

01:14:35
application and let's relaunch it so the moment you relaunch it you can see it is working and okay just want to do one more change it says code compiled in a laptop and in desktop I will say code compiled in desktop okay so you can see we have one computer interface and two classes let's go back here and let's relaunch it and you can see it says code compiled in a laptop that's great so how it is connecting so it is connecting based on the property the property is comp and the reference is laptop and we

01:15:04
got it now what do you think about desktops of course I will talk about desktop in some time but as of now it is working because in the bean tag we have specified laptop laptop since we have laptop here it will work with laptop if you want to use desktop out course you can create a bean for desktop and mention desktop here so I will copy this and this will be desktop and here as well the

class would be desktop whatever you mention here in the reference if you just laptop it will create object of

01:15:31
laptop you can see that it's working for laptop and the moment you set desktop here now it will work for desktop and you can see we got desktop so code compile in a desktop okay so I want to show you a magic here the magic is what if the ID which we are using here for laptop is also com so let's say I'm not specifying as a laptop I'm saying it's a com and here as well I want to use laptop so I will say Comm can we have same name of course we can have right we have seen that before so you can see we

01:16:00
have Comm here and we have Comm here Let's ignore desktop as for time B imagine we don't have any desktop stuff we have a laptop here which whose ID is com and we have com here the moment you run this code you can see it is still working because of course you can have a same name but don't you think when your name and your reference is of same name why to mention that the amazing thing is you can skip it you can just remove this part and you you can say hey doesn't matter what the name is I still want to

01:16:27
get it connected but how so I want to inform spring hey spring when you create object of alien search for the laptop object or search for the computer object you will find it but how so what you will do is in the bean property that you can mention one important aspect which is so let's go back here and say autowire so can you see that we have a concept of autowire and what is autowire now autowire simply means you are informing spring framework hey Spring if you want to set some property if you're

01:16:58
looking for some property if the bean is available connect automatically only when okay only when the bean is available then only connect so you can see it says Auto wire it is searching for the object now so here I will say in fact we have Choice when you say this so if I say control space you can see we have by name or by type so we have two choice so it will try to search a particular element based on the name example here you can see the property name is com so the property name is com and the bean name is also com so it will

01:17:27
directly connect it with the help of Auto Wire by name let me just relaunch it and you can see it is still working so I have not specified any property here but still it is working and that's the beauty of Auto wire trust me auto wire is amazing the moment we start working on Spring boot uh when we start working on GP and other stuff you will love Auto wire okay that works so let me show you one more now let's say I don't have this particular Bean here so I will just commented right so it is commented

01:17:55
now and now you can see we have desktop here so as a alien you need a computer and yes we do have a desktop here so let me just relaunch the application and we got an error the error is because we have not specified the property but you will say hey we have specific autowire right in case of autowire you don't have to mention the property it will search for a particular Bean automatically but unfortunately it was by name by name simply means it will search by name oh that how do we solve it now so what we

01:18:25
will do is instead of going for by name we have another way of achieving this concept now think about this when you say you want the object of alien and alien is depend on what so let's go back to alien and here you can see alien is actually dependent on the computer type so we are not dependent on the name actually we are dependent on the type as well so what if you can go back to Spring XML and you can say hey when you say I want to get object of alien alien is dependent on the object of computer

01:18:56
so what if you can just don't search by name search by type so we can do that as well so here we can say hey search by type now here it will not look for the name example uh if you go back to alien it will not look for com it will look for the object of computer and in this case you can see we have it we have it here so we have the object of desktop with the name desk of course this desktop name is not important what is important is this desktop class implements computer that will be done let's try this so let me just save this

01:19:28
code and let's relaunch the application can you see that we got the output and the output says code compiled in a desktop that's what we want right we wanted it to work and this time I don't want it to search by name I want it to search by type so let me repeat we have two options here autowire can go by name or by type so when you say by name it will try to match the name of the ID or name of the bean when when you say by type it will try to find the type of The Bean that works but then what if you

01:20:00
have this object as well so let's imagine we got two objects one is laptop and one is desktop and when you say by type see of course when you say by name the problem is Sol right of course it will search by name but when you say by type don't you think it will get confused now we have laptop and desktop both so what do you think which one will get preference of course this should get first preference right because this that's a sequence this is the first one there one but then in terms of spring

01:20:25
everything is object right so when you say everything is object of course it should be independent of the sequence and that's where it will get confused the moment you run this code you can see it is giving you error it says no qualifying Bean of type blah blah blah it says I was expecting one but then we found two so your spring framework is getting confused which one to assign is it laptop or desktop how do you solve this that we'll see in the next video how do we solve this error now you can

01:20:59
see we have two candidates here we got laptop object we got desktop object and both are computers and then we are saying hey go by type so that's where your spring is getting confused hey should I go for laptop or desktop now sequence will not be considered now one way to solve this is by going here and going back to the property the moment you have a property here the way we did it before so if you say reference to desktop now this will solve your issue if you want to go with desktop you can

01:21:28
see the moment you mention it will not consider by type or by name it will simply go with the property in fact it will avoid Auto wire so if you run this code you can see problem solved we got

desktop and then if you say comp now this will be giving preference to comp right so problem will be solved so when I say comp it will give preference to laptop so this is one way but what if you don't want to mention this and still you want to go for auto wire by type yes so we have by name we can go with that

01:21:56
or we can mention the property otherwise by type if you want to solve this see even if you have both this objects you have to give preference to at least one and the way you can do that is by using a primary attribute so you can say primary equal to true the moment you do that this particular Bean becomes a primary Bean so in case of confusion it will give preference to this one so example if I on this code now you can see it is giving preference to a laptop so even if you don't have anything

01:22:25
assigned here even if you don't have by name it will go for by type and it will prefer this particular be you can see we got laptop now if you write primary equal to true in desktop it will be getting the first preference right so let's run this code again and you can see we got desktop so that's how you can solve this particular confusion so this is where you can use this Primary in fact the moment you start using annotations so we also have a concept there again we'll use annotation there

01:22:51
so that's about this code we have solved the issue here welcome back aliens my name is ready and in this video we'll talk about spring boot jdbc the thing is if you want to connect your application with database in Java you have to use something called jdbc which stands for Java database connectivity and you will say we know that right so basically you can use jdbc few steps and then you can connect your application with database easily but the problem is as a developer you want to reduce your work right and

01:23:24
also you have to make sure that the application is optimized so let's talk about the optimization part here see a thing is every time you send a request from your application to database it creates a new request and then if you have multiple connections imagine you already have few connections and now you are submitting a new request now you do have this connections right can't we just reuse the same connection instead of getting a new one now that's something you have to manually do or or

01:23:53
you can use something called a data source here now data source will have a configuration and also it provides you something called a connection pooling so basically it will already have few connections and you can use those connections and again you can do this by using any external uh data source or connection pooling concept and you can make it work so that's one thing the second is about developers work so you have to write those beautiful seven or six steps to achieve jdbc and you can do

01:14:35
application and let's relaunch it so the moment you relaunch it you can see it is working and okay just want to do one more change it says code compiled in a laptop and in desktop I will say code compiled in desktop okay so you can see we have one computer interface and two classes let's go back here and let's relaunch it and you can see it says code compiled in a laptop that's great so how it is connecting so it is connecting based on the property the property is comp and the reference is laptop and we

01:15:04
got it now what do you think about desktops of course I will talk about desktop in some time but as of now it is working because in the bean tag we have specified laptop laptop since we have laptop here it will work with laptop if you want to use desktop out course you can create a bean for desktop and mention desktop here so I will copy this and this will be desktop and here as well the class would be desktop whatever you mention here in the reference if you just laptop it will create object of

01:15:31
laptop you can see that it's working for laptop and the moment you set desktop here now it will work for desktop and you can see we got desktop so code compile in a desktop okay so I want to show you a magic here the magic is what if the ID which we are using here for laptop is also com so let's say I'm not specifying as a laptop I'm saying it's a com and here as well I want to use laptop so I will say Comm can we have same name of course we can have right we have seen that before so you can see we

01:16:00
have Comm here and we have Comm here Let's ignore desktop as for time B imagine we don't have any desktop stuff we have a laptop here which whose ID is com and we have com here the moment you run this code you can see it is still working because of course you can have a same name but don't you think when your name and your reference is of same name why to mention that the amazing thing is you can skip it you can just remove this part and you you can say hey doesn't matter what the name is I still want to

01:16:27
get it connected but how so I want to inform spring hey spring when you create object of alien search for the laptop object or search for the computer object you will find it but how so what you will do is in the bean property that you can mention one important aspect which is so let's go back here and say autowire so can you see that we have a concept of autowire and what is autowire now autowire simply means you are informing spring framework hey Spring if you want to set some property if you're

01:16:58
looking for some property if the bean is available connect automatically only when okay only when the bean is available then only connect so you can see it says Auto wire it is searching for the object now so here I will say in fact we have Choice when you say this so if I say control space you can see we have by name or by type so we have two choice so it will try to search a particular element based on the name example here you can see the property name is com so the property name is com and the bean name is also com so it will

01:17:27
directly connect it with the help of Auto Wire by name let me just relaunch it and you can see it is still working so I have not specified any property here but still it is working and that's the beauty of Auto wire trust me auto wire is amazing the moment we start working on Spring boot uh when we start working on GP and other stuff you will love Auto wire okay that works so let me show you one more now let's say I don't have this particular Bean here so I will just commented right so it is commented

01:17:55
now and now you can see we have desktop here so as a alien you need a computer and yes we do

have a desktop here so let me just relaunch the application and we got an error the error is because we have not specified the property but you will say hey we have specific autowire right in case of autowire you don't have to mention the property it will search for a particular Bean automatically but unfortunately it was by name by name simply means it will search by name oh that how do we solve it now so what we

01:18:25
will do is instead of going for by name we have another way of achieving this concept now think about this when you say you want the object of alien and alien is depend on what so let's go back to alien and here you can see alien is actually dependent on the computer type so we are not dependent on the name actually we are dependent on the type as well so what if you can go back to Spring XML and you can say hey when you say I want to get object of alien alien is dependent on the object of computer

01:18:56
so what if you can just don't search by name search by type so we can do that as well so here we can say hey search by type now here it will not look for the name example uh if you go back to alien it will not look for com it will look for the object of computer and in this case you can see we have it we have it here so we have the object of desktop with the name desk of course this desktop name is not important what is important is this desktop class implements computer that will be done let's try this so let me just save this

01:19:28
code and let's relaunch the application can you see that we got the output and the output says code compiled in a desktop that's what we want right we wanted it to work and this time I don't want it to search by name I want it to search by type so let me repeat we have two options here autowire can go by name or by type so when you say by name it will try to match the name of the ID or name of the bean when when you say by type it will try to find the type of The Bean that works but then what if you

01:20:00
have this object as well so let's imagine we got two objects one is laptop and one is desktop and when you say by type see of course when you say by name the problem is Sol right of course it will search by name but when you say by type don't you think it will get confused now we have laptop and desktop both so what do you think which one will get preference of course this should get first preference right because this that's a sequence this is the first one there one but then in terms of spring

01:20:25
everything is object right so when you say everything is object of course it should be independent of the sequence and that's where it will get confused the moment you run this code you can see it is giving you error it says no qualifying Bean of type blah blah blah it says I was expecting one but then we found two so your spring framework is getting confused which one to assign is it laptop or desktop how do you solve this that we'll see in the next video how do we solve this error now you can

01:20:59
see we have two candidates here we got laptop object we got desktop object and both are computers and then we are saying hey go by type so that's where your spring is getting confused hey should I go for laptop or desktop now sequence will not be considered now one way to solve this is by going here and going back to the property the moment you have a property here the way

we did it before so if you say reference to desktop now this will solve your issue if you want to go with desktop you can

01:21:28
see the moment you mention it will not consider by type or by name it will simply go with the property in fact it will avoid Auto wire so if you run this code you can see problem solved we got desktop and then if you say comp now this will be giving preference to comp right so problem will be solved so when I say comp it will give preference to laptop so this is one way but what if you don't want to mention this and still you want to go for auto wire by type yes so we have by name we can go with that

01:21:56
or we can mention the property otherwise by type if you want to solve this see even if you have both this objects you have to give preference to at least one and the way you can do that is by using a primary attribute so you can say primary equal to true the moment you do that this particular Bean becomes a primary Bean so in case of confusion it will give preference to this one so example if I on this code now you can see it is giving preference to a laptop so even if you don't have anything

01:22:25
assigned here even if you don't have by name it will go for by type and it will prefer this particular be you can see we got laptop now if you write primary equal to true in desktop it will be getting the first preference right so let's run this code again and you can see we got desktop so that's how you can solve this particular confusion so this is where you can use this Primary in fact the moment you start using annotations so we also have a concept there again we'll use annotation there

01:22:51
so that's about this code we have solved the issue here welcome back aliens my name is ready and in this video we'll talk about spring boot jdbc the thing is if you want to connect your application with database in Java you have to use something called jdbc which stands for Java database connectivity and you will say we know that right so basically you can use jdbc few steps and then you can connect your application with database easily but the problem is as a developer you want to reduce your work right and

01:23:24
also you have to make sure that the application is optimized so let's talk about the optimization part here see a thing is every time you send a request from your application to database it creates a new request and then if you have multiple connections imagine you already have few connections and now you are submitting a new request now you do have this connections right can't we just reuse the same connection instead of getting a new one now that's something you have to manually do or or

01:23:53
you can use something called a data source here now data source will have a configuration and also it provides you something called a connection pooling so basically it will already have few connections and you can use those connections and again you can do this by using any external uh data source or connection pooling concept and you can make it work so that's one thing the second is about developers work so you have to write those beautiful seven or six steps to achieve jdbc and you can do

01:24:22

that but the only problem is the more code you write you are basically increasing the number of lines which will have a bug so you know every new line is a potential bug right so reduce the number of lines which you write and use certain tools of Frameworks which will do it for you and these Frameworks and tools are been tested by everyone so if it works it works for you as well now that's where we have to use a template now there's something called jdbc template now jdbc template will help you

01:24:50
with data source that means the connection pulling concept will be done and it will also help you to write your code in an efficient way and all these things are provided in Spring boot jdbc now why spring boot is because if you want to build Enterprise application and if you want to follow all the design patterns best practices and a framework where you can get the libraries and everything working together spring is one of the best framework and that's why we're going to use spring boot jdbc is

01:25:15
because under the spring umbrella we have lot of projects and this is one of it so let's get started with the code where we going talk about how do you build an application and we'll learn by building something right so it will make much more sense so let's start the code the thing is if you want to build this project you need a IDE now you can choose any ID which you want maybe you can use Eclipse intellig vs code your choice I'm going to use intell idea and that to a community version because

01:25:44
that's for free and it works so we'll be using that so to get the project for spring if you're using a spring tools on vs code or Eclipse you can directly get the project from there and if you're using the ultimate version of intell idea you can create the project directly from your IDE but since we are using a community version I'm going to use the spring inst viser so you can head to the start. spring.io and you can get your project from here the base skeleton of the project so the first thing you will

01:26:15
mention is a project type I'm going to go with mavin uh language I will choose Java of course that's what you're doing now and then the spring boot version I'm selecting 3.2 next you have to mention the group ID so I'm going to select I'm going to type here com. telescope the project since I'm working with spring jdbc demo that's the project name I I will give and will not change anything here the packaging I want it to be type of jar and the version for Java I'm

01:26:44
using in my machine is 21 now based on which version you have you can either just choose 21 or you can select 17 uh dependencies so basically we have to add uh two things only one is the jdbc API now this is your springboard jdbc and you have to add a database now which database we going to use of course you can choose any dbms which you want now we will be using a embedded one so if you have post msql Oracle mssql you can use that the only thing is you have to do some additional configuration and

01:27:18
I will tell you where to do that configuration but we are going to use a inmemory database here and by default when you use inmemory like hed2 it will be pre-configured so you don't have to do any configuration for that so this these are the two libraries we need and that's it you can click on generate it will give you a project so as you can see I got the project downloaded which is spring

jdbc demo I will just unzip it and open in any ID which you want I'm going to use intj so let me do that

01:27:46
quickly okay so basically have opened that project in my IDE and it will take some time for it to resolve the dependencies now we have added two which is jdbc and the hed2 and we don't have to manually download the H2 driver because the moment you add H2 it will also have a driver installed so if I open the pal. XML you can see we do have H2 here and we got spring boot starter jdpc this is what we want and now we can get the work started so we'll go back here to main Java and spring jdbc demo

01:28:20
application and we can start the coding here okay so one thing I mentioned before right if you want to use P or Mas or any other dbms variable to the configuration so if you can see we have a resources folder here and if you open application. properties this is where you have to specify those properties so what properties I'm talking about you have to specify the URL you have to specify the username and password and Driver now basically this variable belongs to spring. dat source. URL and you can set the value for the URL

01:28:51
whichever you want so that's how you do it but now we are not going to use Mas or pus we are going to use H2 which is in memory reason I'm doing using H2 here is because you can directly take this project and run on your machine if I use P or MySQL then you have to also install those softwares you have to configure it and there will be some issues with that so let's go stick to H2 okay so what we want to do we want to basically store some data in the database and fetch it from database so we do have database

01:29:20
ready so the moment you run this application you will be having Edge to database behind the scene running for you but then what you want to store so in that case what I will do is I will create a simple class which will represents a particular table so normally you can have a relation between a table and your class now you will say how is is it possible see the thing is if you think about a table table will have a table name table will have a columns let's say we are working with aliens so every

01:29:48
programmer is alien right so a programmer or alien need an ID a alien needs a name and a alien needs a favorite technology so let's say for me my favorite technology is Java blockchain AI for you it might be something different so basically every alien will need a technology as well so three columns in the table and then this columns or this table will also have some data let's say one1 which is ID and naven Technology Java let's say two Gan technology blockchain so basically you can have this data in the same way if if

01:30:22
you try to relate this with a class now in a class basically what you'll have is you will have a class of course a class will have a name a class will have properties right variables now if you try to match the class name and the table name look similar properties of this class which is ID name and Tech matches with the column names right and the objects for this alien class will be the rows each object will be one row now that's how basically you map between the class and the object and then if you if....give important and teach me like this only in very very simple

Alright Samiksha 😊
You gave a **huge** text, but don't worry — I will **pull out ONLY the important concepts** and explain them in **super simple language**, just like you want.

We will go in **small blocks**, very slow, very clear, exactly like Navin Sir but easier.

🌟 PART 1 — Reference Type Injection (Laptop inside Alien)

We already did:

- Age = **primitive** → use **`value`**
- Laptop = **object** → cannot use **`value`**

Now we add:

```java
Laptop laptop;
```

and we got **`NullPointerException`** because:

> We never gave Alien a laptop object!

Solution:

Step 1: Create Bean for Laptop

```xml
<bean id="laptop" class="com.test.Laptop" />
```

You MUST create a bean for every class you want Spring to manage.

Step 2: Inject Laptop into Alien

In Alien bean:

```xml
<bean id="alien" class="com.test.Alien">
    <property name="laptop" ref="laptop"/>
</bean>
```

Meaning:

- **`property name="laptop"`** → Alien has variable laptop
- **`ref="laptop"`** → Use the bean with ID "laptop"

💡 **'value'** is for:

- int
- float
- String
- boolean

💡 **'ref'** is for:

- Objects / classes

🌟 PART 2 — Constructor Injection (very simple)

Until now we used:

```xml
<property name="age" value="10"/>
```

This is **setter injection**.

But we can also set values using **constructor**.

Alien:

```java
public Alien(int age) {
    this.age = age;
}
```

Spring XML:

```xml
<bean id="alien" class="com.test.Alien">
    <constructor-arg value="12"/>
</bean>
```

Important rule:

❗ If you remove default constructor and DON'T specify constructor-arg:
→ Spring throws error: "No default constructor".

🌟 PART 3 — Constructor Injection for Object Reference

If constructor takes:

```java
public Alien(int age, Laptop laptop) { ... }
```

Then XML becomes:

```xml
<bean id="alien" class="com.test.Alien">
    <constructor-arg value="12"/>
    <constructor-arg ref="laptop"/>
</bean>
```

Small summary:

- `<property>` → Setter Injection
- `<constructor-arg>` → Constructor Injection
- Use **constructor** when variable is *mandatory*
- Use **setter** when variable is *optional*

🌟 PART 4 — Interface Injection (Computer instead of Laptop)

Alien should not depend on laptop only.
Better to depend on **Computer**.

Step:

Make interface:

```java
interface Computer {
    void compile();
}
```

Laptop implements it:

```java
class Laptop implements Computer {
    public void compile() {
        System.out.println("Compiled in Laptop");
    }
}
```

Desktop implements it:

```java
class Desktop implements Computer {
    public void compile() {
        System.out.println("Compiled in Desktop");
    }
}
```

Alien now does:

```java
Computer comp;
comp.compile();
```

XML:

```xml
<bean id="laptop" class="com.test.Laptop" />
<bean id="desktop" class="com.test.Desktop" />

<bean id="alien" class="com.test.Alien">
    <property name="comp" ref="laptop"/>
</bean>
```

To use desktop:

```xml
<property name="comp" ref="desktop"/>
```

🌟 PART 5 — Autowiring (Magic 🔥)

If bean name = property name, we can skip `<property ref="">`

Example:

Alien has:

```java
Computer comp;
```

Laptop bean id is also:

```xml
<bean id="comp" class="Laptop"/>
```

Then:

```xml
<bean id="alien" class="Alien" autowire="byName"/>
```

Spring says:

> Property name = comp
> Bean name = comp
> Automatically connect ✔️

No need to write:

```xml
<property name="comp" ref="comp"/>
```

🌟 PART 6 — Autowire byType

If you write:

```xml
<bean id="alien" class="Alien" autowire="byType"/>
```

Spring will search:

> "Give me any bean of type Computer"

If only **one** class implements Computer → works.

If TWO classes exist (Laptop + Desktop) → Spring becomes confused:

❌ Error: "No qualifying bean — expected 1, found 2"

🌟 PART 7 — Solving “Multiple Candidate Beans” Problem

We have:

- Laptop implements Computer
- Desktop implements Computer
- Autowire byType → conflict

Two solutions:

✔️ Solution 1: Use property manually

```xml
<property name="comp" ref="laptop"/>
```

Then Spring ignores autowire.

✔️ Solution 2: Use `primary="true"`

Laptop:

```xml
<bean id="laptop" class="Laptop" primary="true"/>
```

Now when autowiring by type:

> Laptop will be chosen automatically.

If desktop should win:

```xml
<bean id="desktop" class="Desktop" primary="true"/>
```

🌟 PART 8 — Why Autowire is Super Important?

You saw:

- byName
- byType
- Primary

In Spring Boot we will NEVER write XML.
Instead we use:

```java
@Autowired
private Computer comp;
```

Spring Boot uses **byType + primary** in the background.

That's why this part is very important to understand.

🌟 PART 9 — Spring Boot JDBC (simple explanation)

Before JDBC template:

- You write 7 steps
- Open connection every time → slow
- More code → more bugs

Spring gives **JdbcTemplate**, which:

- Uses DataSource
- Automatically manages connections (pooling)
- Makes JDBC code super short

Boot handles:

- Driver loading
- DB URL

- Username/password
- Connection pool
- JdbcTemplate bean creation

You only write queries.

🌟 FINAL SUPER SHORT SUMMARY

Concept	Super Simple Meaning
primitive property	value=""
reference property	ref=""
setter injection	**`<property>`**
constructor injection	**`<constructor-arg>`**
interface	flexibility
autowire byName	match variable name to bean id
autowire byType	match variable type to bean type
primary="true"	in case of confusion choose this bean
BeanFactory	old container
ApplicationContext	new, full-featured container
Singleton	object created once
Prototype	new object every getBean()
JdbcTemplate	easy JDBC with connection pooling

🌟 If you want, next I can explain:

✔️ “Annotations version of all this (no XML)”
✔️ “@Component, @Autowired, @Primary, @Qualifier – simple explanation”
✔️ “Spring Boot version of Alien–Laptop example”

Just say the topic.

programmer is alien right so a programmer or alien need an ID a alien needs a name and a alien needs a favorite technology so let's say for me my favorite technology is Java blockchain AI for you it might be something different so basically every alien will need a technology as well so three columns in the table and then this columns or this table will also have some data let's say one1 which is ID and naven Technology Java let's say two Gan technology blockchain so basically you can have this data in the same way if if

01:30:22
you try to relate this with a class now in a class basically what you'll have is you will have a class of course a class will have a name a class will have properties right variables now if you try to match

the class name and the table name look similar properties of this class which is ID name and Tech matches with the column names right and the objects for this alien class will be the rows each object will be one row now that's how basically you map between the class and the object and then if you if

01:30:52
you're using any springing orm technology let's say hibernate in that case the table creation happens automatically now since we're not using hibernate here we have to create the table by ourself yeah that's something you have to do okay let's create a class here and I want to create a class for the let's say alien okay so when I say alien class what I want to do is I want to basically create few variables so I will say private in ID and then private string name and private string

01:31:22
Tech so basically I need these three variables and for these three variables I also need Getters and Setters so we'll right click here and we'll say generate Getters and Setters I will just choose these three variables done I also need a two string method if I want to print the data itself so I will just right click here say generate uh two string method and done I will do one more thing I will move this particular class the alien class to a different package so this should be into a a package of model so every class

01:31:54
should belong to their own packages so alien is a model class so I will just move that to a model package and since we don't have a model package I will say hey move to model package in the SRC yes and you can see it is moved now so that's done but if I want my spring framework to manage this particular bean I have to use add component now if you're not sure how spring boot Works uh you you can check out the video link in the description I will make sure that you have your spring boot video there

01:32:21
okay so basic basically what we are doing with ADD component is we are asking spring framework to handle this object creation object managing and object destroying by Spring framework I don't want to do that okay so basically when I say I don't have to create the object how spring will do it so basically what I will do is uh this run method so I will go back to the main class so main class is your spring jdbc demo application and in this the you can see we are calling a run method run Returns the object of application

01:32:48
context so I will say application context cont text now why I'm doing this is because I want to get the hold on a spring container again I'm assuming that you know spring boot here because if I start teaching your spring board then it will take it will make the video length of five hours I don't want to do that so again you'll find the link in description you can check it out now let me create the object for alien so I will say alien I will say alien one is equal to okay we have to basically import the

01:33:12
package for alien which is in the model package done and now I want to create the object the way you can create the object by saying alien right but I don't want to create this object by myself I just want this to be done by Spring framework so I will say context. get bean I want the object for alien do class right uh but the thing is by default the type of this alien component is Singleton so basically the ban is Singleton I want to make sure this is prototype so I will say scope prototype

01:33:41
so what will happen is for different request it will create new objects for you otherwise you will have multiple alien object or multiple alien references with the same object so you will have the same data doesn't matter how many data you have in your table okay so that's done and for this alien what I will do is I will add some values so let's say the ID I want to add is 111 and I want to say not get ID I want to say set ID alien do set name I want to set the name as let's say naven and

01:34:10
alien dot alien 1. set Tech I want to set the tech as let's say Java now basically I'm adding this values and now I got my object ready I want to store this object in database how will I do that maybe I can simply say save this data in database of course this will not work so what you have to do and you know why this will not work is because you have not subscribed the channel yet so if this works on your machine that means you have already subscribed okay I'm just kidding so this will not work right

01:34:41
we have to write Java code there now how will you make this work now basically this is where you have to write extra classes see what happens is in normal application you'll be having multiple layers you'll be having a controller so if you're building a web application you'll be having a controller layer you'll be having a service layer you'll be having a repository layer now this repository layer is responsible to connect with the database of course I can write my jdbc steps here in the main

01:35:04
because that's a normal main method I can write all the beautiful jdbc steps here but then we don't want to do that that's why we are doing springboard jdbc here and this is not a good idea to write your JBC code inside main so this should be a part of a separate file now instead of creating a service layer I will be skipping that and I will directly jump into a depository layer so I will just create here a depository layer so I will say new class and I will this I will say this is a alien repo or

01:35:34
you can also say da if you want now repo is responsible or repository is responsible to connect with database okay now this is where you will do this stuff but first of all uh we need to have some methods here maybe I want to have a method like public void save so it will save the object Which object the alien object alien alien into the database okay so we have to import the package I will also make sure that this belongs to a service or the repo package move to repo yeah okay so here what I will do is I will say at this point I'm

01:36:10
not writing all the codes I will say added and then I also need a method which will return the least of aliens so let's say if I want to fetch all the aliens from the database now in this case I have I need a method called maybe find all method name you can change as you want I'm just saying find all here import the package and here let me return a new add list of aliens okay and I'm I'm just reading an empty object but we'll we'll code this in some time the idea is I want this to work first of all

01:36:42
and since I want spring to manage it I will write add component so instead of adding add component what makes sense is to use add repository for for different layers we have different stereotype annotations for repository use repository go back here now if you want to store this data in the database I need object of alien repo I will say repo is equal to instead of saying new

repo or new alien repo we will be using context. get bean the alien repo do class this is I want now once we got the object I can simply say repo dot I want

01:37:17
to save the object let's say I want to save alien object alien one and maybe I want to print print all the objects so get all the objects from the table or all this all the aliens from the table and print it here we have not done any changes yet and we are not written any code to connect database yet but let's see if this works right click here and say run okay taking some time because I'm running this for the first time and you can see it says uh empty why is not calling oh I said added I wanted to

01:37:47
print added what's wrong with me okay so let's run this and now now you know why it has not worked before why I said SL slash added is because I realized you don't like the videos for me anyway so you can see we got added here and then we got the square brackets which is empty basically empty at least now let's make this work so basically we have to make sure that we write a code in the alien repo this is the only file you have to work with now we have to write a code where we can store this data in the

01:38:19
database so how exactly we going do that it's very simple basically we have to use a jdbc template now this is a part of your spring boot jdbc so what you to do is you have to get hold on there something called jdbc template you just have to create the object of this I will say this is template okay so you have to get hold on the jdbc template for this this is a property right and this is the inbuilt stuff if you can see jdbc template belongs to springf framework. jdbc Doc okay and and you will see how

01:38:48
easy it is to connect and all the all the stuff right you don't have to write right connection con driver manager remember in jdbc we do that and we don't have to prepare this statement all this thing will be happening behind the scene but for this I need data sets I can just right click here say generate data sets for this JB template that's done and I want this object to be injected so because see we are just creting a reference of it right where is the object so spring will take care of

01:39:14
it you don't have to worry because if you go to JBC template we can create object of this class but again for that you have to do all the configuration you can see we have to pass the object of object of data source I don't want to do that I just want my spring framework to manage it and the way you can do that is by just writing autowire your job is done you got the jdbc template object Now using that object let's say I want to I want to add this I want to add the alien object it's very simple now you

01:39:43
can simply use the template now this template which is a jdbc template object you have multiple methods now in Normal DBC if you want to execute a statement now based on which statement it is is it a select query or is it other queries like delete update now if you working with select query which is the reading you basically use execute query as a method if you want to change something let's say update insert delete you use execute update here we want to save it right so we'll be using execute update

01:40:15

execute update takes two parameters in fact multiple parameters depend upon how you're doing it the first one is the query of course you need a query with you so I say string St Str or string SQL is equal to what is a query now here I'm going to use a query like insert into what is my table name nowhere we have mentioned a table name right let me make it similar to the class name let's say alien and then how many fields I have so I have let's say ID I got name and I got Tech okay and then I want to specify

01:40:48
values as well now where are the values so you tell me where are the values values are here inside this alien object you got the values we just have to pass this values here now if you remember the prepared statement we don't directly pass the values what we do is we put question marks there so that we can replace this question mark later so we are using repare statement by default just pass the query here or pass the SQL here after passing the SQL you have to also replace this question marks with

01:41:14
the actual values so those are the remaining parameters you have so after mentioning the query just say alien do get ID comma alien. get name comma alien dog Tech that's it this will basically save it but will it work that's a question right so let's see what happens if you run this and we'll try to solve that later right click here and say run I'm expecting error but let's see what errors you get okay we got the error and the error is very simple it says the table alien not found of course

01:41:53
right H2 will give you database not the table table is something you have to create by yourself now how will I create the table and that's where we have to add a schema in the resource folder so I have to add two things actually one is schema for the structure and what if you want some default values maybe you know every time you create a table you want certain values there to test it for that you have to create a file called data tosql and we'll do that in this so let's make it work so what we are missing here

01:42:21
is we are missing this table and we have I told solution we have to create a schema file because hed2 when it loads we don't have a table there right now to create that file it's very simple you can right click on resources and say new and create a file called schema.sql and say enter now this is where you will write your script now the script to create your table so the script for creating a table or the SQL query is create table and we'll give the table name as alien and then you have to

01:42:52
mention the properties or the the the columns so the column which I want here is the first one is ID now the ID will be of type uh int and we'll say this is a primary key uh then you'll be having a name name will be of type Vare and I will say the Vare is 20 in fact uh you know maybe we have done this before so normally in 20 characters you will name will be there but then if you talk about South indan names I'm a South so I have a very big name no na is not my full name my full name is tumar B I know

01:43:26
that's huge so maybe I will need 50 characters and then here I will say marks and again marks will be of type in and here we'll be having Tech and Tech will be of again of Ware and I will say Ware 20 because Technologies can be shorten up if you even if you say artificial intelligence it is AI so let's keep it there so basically this is the file for the schema now we also need a file for the data again not compulsory but then I want to have some data so I will have some insert statement here so

01:43:57
I will say insert into alen and I want to insert three columns ID name and Tech and what are the values I'm going to have here so I will say value is one1 then name in single code let's say kid and let's say technology is blockchain that's one and maybe I want two more so I will just copy and paste it here so this is one2 and this is hsh and let technology is AI 103 and this is social let's say iot okay so basically we got this uh three rows already created and now if you run this code so

01:34:10
alien dot alien 1. set Tech I want to set the tech as let's say Java now basically I'm adding this values and now I got my object ready I want to store this object in database how will I do that maybe I can simply say save this data in database of course this will not work so what you have to do and you know why this will not work is because you have not subscribed the channel yet so if this works on your machine that means you have already subscribed okay I'm just kidding so this will not work right

01:34:41
we have to write Java code there now how will you make this work now basically this is where you have to write extra classes see what happens is in normal application you'll be having multiple layers you'll be having a controller so if you're building a web application you'll be having a controller layer you'll be having a service layer you'll be having a repository layer now this repository layer is responsible to connect with the database of course I can write my jdbc steps here in the main

01:35:04
because that's a normal main method I can write all the beautiful jdbc steps here but then we don't want to do that that's why we are doing springboard jdbc here and this is not a good idea to write your JBC code inside main so this should be a part of a separate file now instead of creating a service layer I will be skipping that and I will directly jump into a depository layer so I will just create here a depository layer so I will say new class and I will this I will say this is a alien repo or

01:35:34
you can also say da if you want now repo is responsible or repository is responsible to connect with database okay now this is where you will do this stuff but first of all uh we need to have some methods here maybe I want to have a method like public void save so it will save the object Which object the alien object alien alien into the database okay so we have to import the package I will also make sure that this belongs to a service or the repo package move to repo yeah okay so here what I will do is I will say at this point I'm

01:36:10
not writing all the codes I will say added and then I also need a method which will return the least of aliens so let's say if I want to fetch all the aliens from the database now in this case I have I need a method called maybe find all method name you can change as you want I'm just saying find all here import the package and here let me return a new add list of aliens okay and I'm I'm just reading an empty object but we'll we'll code this in some time the idea is I want this to work first of all

01:36:42
and since I want spring to manage it I will write add component so instead of adding add component what makes sense is to use add repository for for different layers we have different stereotype annotations for repository use repository go back here now if you want to store this

data in the database I need object of alien repo I will say repo is equal to instead of saying new repo or new alien repo we will be using context. get bean the alien repo do class this is I want now once we got the object I can simply say repo dot I want

01:37:17
to save the object let's say I want to save alien object alien one and maybe I want to print print all the objects so get all the objects from the table or all this all the aliens from the table and print it here we have not done any changes yet and we are not written any code to connect database yet but let's see if this works right click here and say run okay taking some time because I'm running this for the first time and you can see it says uh empty why is not calling oh I said added I wanted to

01:37:47
print added what's wrong with me okay so let's run this and now now you know why it has not worked before why I said SL slash added is because I realized you don't like the videos for me anyway so you can see we got added here and then we got the square brackets which is empty basically empty at least now let's make this work so basically we have to make sure that we write a code in the alien repo this is the only file you have to work with now we have to write a code where we can store this data in the

01:38:19
database so how exactly we going do that it's very simple basically we have to use a jdbc template now this is a part of your spring boot jdbc so what you to do is you have to get hold on there something called jdbc template you just have to create the object of this I will say this is template okay so you have to get hold on the jdbc template for this this is a property right and this is the inbuilt stuff if you can see jdbc template belongs to springf framework. jdbc Doc okay and and you will see how

01:38:48
easy it is to connect and all the all the stuff right you don't have to write right connection con driver manager remember in jdbc we do that and we don't have to prepare this statement all this thing will be happening behind the scene but for this I need data sets I can just right click here say generate data sets for this JB template that's done and I want this object to be injected so because see we are just creting a reference of it right where is the object so spring will take care of

01:39:14
it you don't have to worry because if you go to JBC template we can create object of this class but again for that you have to do all the configuration you can see we have to pass the object of object of data source I don't want to do that I just want my spring framework to manage it and the way you can do that is by just writing autowire your job is done you got the jdbc template object Now using that object let's say I want to I want to add this I want to add the alien object it's very simple now you

01:39:43
can simply use the template now this template which is a jdbc template object you have multiple methods now in Normal DBC if you want to execute a statement now based on which statement it is is it a select query or is it other queries like delete update now if you working with select query which is the reading you basically use execute query as a method if you want to change something let's say update insert delete you use execute update here we want to save it right so we'll be using execute update

01:40:15
execute update takes two parameters in fact multiple parameters depend upon how you're doing it the first one is the query of course you need a query with you so I say string St Str or string SQL is equal to what is a query now here I'm going to use a query like insert into what is my table name nowhere we have mentioned a table name right let me make it similar to the class name let's say alien and then how many fields I have so I have let's say ID I got name and I got Tech okay and then I want to specify

01:40:48
values as well now where are the values so you tell me where are the values values are here inside this alien object you got the values we just have to pass this values here now if you remember the prepared statement we don't directly pass the values what we do is we put question marks there so that we can replace this question mark later so we are using repare statement by default just pass the query here or pass the SQL here after passing the SQL you have to also replace this question marks with

01:41:14
the actual values so those are the remaining parameters you have so after mentioning the query just say alien do get ID comma alien. get name comma alien dog Tech that's it this will basically save it but will it work that's a question right so let's see what happens if you run this and we'll try to solve that later right click here and say run I'm expecting error but let's see what errors you get okay we got the error and the error is very simple it says the table alien not found of course

01:41:53
right H2 will give you database not the table table is something you have to create by yourself now how will I create the table and that's where we have to add a schema in the resource folder so I have to add two things actually one is schema for the structure and what if you want some default values maybe you know every time you create a table you want certain values there to test it for that you have to create a file called data tosql and we'll do that in this so let's make it work so what we are missing here

01:42:21
is we are missing this table and we have I told solution we have to create a schema file because hed2 when it loads we don't have a table there right now to create that file it's very simple you can right click on resources and say new and create a file called schema.sql and say enter now this is where you will write your script now the script to create your table so the script for creating a table or the SQL query is create table and we'll give the table name as alien and then you have to

01:42:52
mention the properties or the the the columns so the column which I want here is the first one is ID now the ID will be of type uh int and we'll say this is a primary key uh then you'll be having a name name will be of type Vare and I will say the Vare is 20 in fact uh you know maybe we have done this before so normally in 20 characters you will name will be there but then if you talk about South indan names I'm a South so I have a very big name no na is not my full name my full name is tumar B I know

01:43:26
that's huge so maybe I will need 50 characters and then here I will say marks and again marks will be of type in and here we'll be having Tech and Tech will be of again of Ware and I will say Ware 20 because Technologies can be shorten up if you even if you say artificial intelligence it is AI so let's keep it there so basically this is the file for the schema now we also need a file for the data again

not compulsory but then I want to have some data so I will have some insert statement here so

01:43:57
I will say insert into alen and I want to insert three columns ID name and Tech and what are the values I'm going to have here so I will say value is one1 then name in single code let's say kid and let's say technology is blockchain that's one and maybe I want two more so I will just copy and paste it here so this is one2 and this is hsh and let technology is AI 103 and this is social let's say iot okay so basically we got this uh three rows already created and now if you run this code so

01:44:45
we have created two files schema and data SQL I will just go back here to my main application and run this this and let's see what happens now so when you run this let's see in the taking some time again okay so you can see there's no problem there's no error and we got uh empty here so so the reason it got empty is because if you see the repo we are still ring the empty data so we are not ring data from the table okay so the data is created how do I know that data is created what I will do is uh you know

01:45:16
this update returns you the number of rows affected so you can see we got int so what I will do here is I will say rows and I want to see how many rows are affected and then I will print the number of rows affected so I will say rows plus space rows affected okay so it will print how many rows got affected now if I run this code of course we'll still the empty at list but at least I want to see is it getting stored in the database that's important so it says one row or rows affected here that's good so that means

01:45:48
it is we are able to insert but now I want to fetch it how do I fetch it now to fetch the data again we can use our jdbc template because if you see the template dot there's one more method here which is called query now query returns you the data from table right so I will say query but what is my query so I will say string SQL basically I want to fetch all the records I will say select star from Alien so what it will do is it will return all the rows and then okay let's see what happens so if I

01:46:19
say SQL it will return me the rows right but in which format now basically the query you can see we have multiple methods of query to fetch the data so we can use string which is the SQL and then you can use result set extractor or you can say row call back Handler or you can use we can use Row mapper for our example row mapper makes sense now what rowmapper will do is I I will show you one more time so we have to basically pass the object of row mapper so what happens is when you get all the data in

01:46:53
your result set you want to fetch one row at a time right so row mapper will help you to give you one row at a time to do that we have to create the object of row mapper so let's create one so I will say row mapper and row mapper will be working with a type alien let's say this is mapper equal to now how do we create object of row mapper so we can say new row mapper but the problem is if you go to row mapper it is a interface this is a function interface which has only one method called map row and which

01:47:24
takes two parameters one is a result set now this result set will be given by your query and then it also gives you a row number so in result set you might be having multiple rows right and you want

to work with one row at a time so this row number will take care of it okay so basically I have to create object of row mapper which has a method called map row and then I have two things here which is this set and row number now basically this is the interface right so we have to create a Anonymous in a class to do

01:47:56
that or we can also create a separate class which implements row mapper then create the object of it again that's your choice but here I'm using Anonymous in a class and slowly we'll convert that into Lambda as well so that it will be shter code but let's say we have this object here which is mapper and this is our Anonymous in our class so basically uh we have to do something here so what we have to do is this result set will have all the records right we have to take one record time so what map row

01:48:22
does is it takes the entire result set and it executes map R multiple times depend upon how many rows you have so if you have four rows the map row will execute four times and it will give you one object at a time so basically we'll take that we'll create object of alien here I will say alien a is equal to new alien and then this alien object is empty right as of now and I've written the same object so you can see row mapper needs object of alien so I'm getting the object here and then I'm

01:48:54
returning it but then this is an empty object right so it will not have values the values are there in result set so I can simply say a do set ID and now get the value from result set so you can say get int so how do you get data from result set you have to say get int and pass the column number so either you can say column number or you can say column name both work so here you can also mention ID a do set name this will be coming from the second row so get string I will say two here and then a do

01:49:25
set Tech RS do get string three as I mentioned before you can also use a number or you can use a name of the column and that's it once you got the data from result set you are giving it to the alien object and then you are returning it so basically R mapper will take care of the data one by one now the same object which is of mapper you have to pass it here so you're not just passing a query you are also specifying what to do with the result set so if you click on query it returns you the object

01:49:59
of list of aliens so the type is T alien so we have to accept the list of alien I will say aliens is equal to this so basically this query will return the list of aliens and that's what you are going to return not the new at least so what we did is we fired the query we got the data it's just that we have to map it we have to convert that data from result set into the list of aliens and that's what we have done with the help of mapper and that's it I hope this will work let's see let me remove some extra

01:50:31
spaces there so with this if I run this code and again I hope it will work let's run okay can you see that it worked so basically alien 101 name kidan technology blockchain 102 H AI 103 suil iot and then the new entry which we have added 111 naven and Java this is what we have added right so basically these things are working out that's how you work with the spring boot with your jdbc and yeah that's it in fact you can do some more changes here if you want uh since this is a function interface we

01:51:05
can use Lambda expression here so we can remove from here to here cut we can also remove one cly brackets here and you have you just have to put the arrow now let's remove the SQL exception let's put the arrow and then in Lambda we don't even need the type so we can remove both the types here and you know one more thing the row number is not not getting used right so we can even use R there or any other variable if you want let's say a again it will not make it readable right because a is already there let's

01:51:36
say row number or we can say okay so this is the shorter code now you can also replace this mapper with this code again this is optional if you want to do it you can try to use Lambda not compulsory so we don't need to create the the reference here we can directly use this right so that's how we can do it in fact you can also remove this part you can directly write this particular statement in return even that works so if I run this code once again so you can see it works so yeah that's your spring

01:52:06
Boot jdbc and if you observed everything is happening with the help of jdbc template so you don't have to basically do all this configuration by yourself writing the connection statement creating the statement everything will be taken care by the template you just have to specify the query and fire the [Music] query welcome back everyone in this module we'll talk about spring MVC now when you create a spring MVC project there are two ways of doing it so in the earlier days we were having only one way

01:52:36
which was spring MVC configuration but now we also got spring boot so the idea of spring boot is just to make thing easy okay so in Spring MVC as well we can achieve all the thing but just that we have to do a lot of configuration and to solve that we have spring boot okay but what is spring MVC to understand spring MVC we have to understand what is MVC and to understand MVC we have to understand why we have MVC so let's go back to the old days when you talk about normal servit and JSP all let's talk

01:53:06
about servlet so whenever a client sends the request to the server it is server's responsibility to process the request and give it back to the client but as a client what a client will get so of course client need two things the first thing is client needs data of course right if you go to any website the first thing you need is data but you want that data in a well formatted way example when you go to Facebook on Facebook you see data plus you also see a proper layout of course every one will get the

01:53:37
same layout so if I go to Facebook and you go to Facebook we'll be seeing the same screen the same colors right what will change is data so your friend list different from my friend list your feeds will be different from my feeds what will remain common is design right so we thought as a s instead of creating everything because subet on the server side will create the layout and will create data yes you can do that but you know it becomes clumsy we feel that you know you should be have a separate task

01:54:06
or we we should be having separate things for separate task example we need someone who will accept the request we need someone who will give the design okay uh it's like a normal template

so design will remain common what will changes data and to hold that data we want something so basically we want something which will accept the request we want something which will be viewed to the client and we want something where it will hold the data so we got a controller who will accept the request

01:54:35
who will send the response we want a view technology and then we want a model so model basically will hold the data and that's where we got this concept of MVC model view controller so if you have model view controller it will almost work same way as normal seret without using MVC but then the advantage would be we are separating the concerns okay we are separating the task so the controller will only accept the request in the earlier days we used to use salet for controllers and JSP for View and

01:44:45
we have created two files schema and data SQL I will just go back here to my main application and run this this and let's see what happens now so when you run this let's see in the taking some time again okay so you can see there's no problem there's no error and we got uh empty here so so the reason it got empty is because if you see the repo we are still ring the empty data so we are not ring data from the table okay so the data is created how do I know that data is created what I will do is uh you know

01:45:16
this update returns you the number of rows affected so you can see we got int so what I will do here is I will say rows and I want to see how many rows are affected and then I will print the number of rows affected so I will say rows plus space rows affected okay so it will print how many rows got affected now if I run this code of course we'll still the empty at list but at least I want to see is it getting stored in the database that's important so it says one row or rows affected here that's good so that means

01:45:48
it is we are able to insert but now I want to fetch it how do I fetch it now to fetch the data again we can use our jdbc template because if you see the template dot there's one more method here which is called query now query returns you the data from table right so I will say query but what is my query so I will say string SQL basically I want to fetch all the records I will say select star from Alien so what it will do is it will return all the rows and then okay let's see what happens so if I

01:46:19
say SQL it will return me the rows right but in which format now basically the query you can see we have multiple methods of query to fetch the data so we can use string which is the SQL and then you can use result set extractor or you can say row call back Handler or you can use we can use Row mapper for our example row mapper makes sense now what rowmapper will do is I I will show you one more time so we have to basically pass the object of row mapper so what happens is when you get all the data in

01:46:53
your result set you want to fetch one row at a time right so row mapper will help you to give you one row at a time to do that we have to create the object of row mapper so let's create one so I will say row mapper and row mapper will be working with a type alien let's say this is mapper equal to now how do we create object of row mapper so we can say new row mapper but the problem is if you go to row mapper it is a interface this is a function interface which has only one method called map row and which

01:47:24
takes two parameters one is a result set now this result set will be given by your query and then it also gives you a row number so in result set you might be having multiple rows right and you want to work with one row at a time so this row number will take care of it okay so basically I have to create object of row mapper which has a method called map row and then I have two things here which is this set and row number now basically this is the interface right so we have to create a Anonymous in a class to do

01:47:56
that or we can also create a separate class which implements row mapper then create the object of it again that's your choice but here I'm using Anonymous in a class and slowly we'll convert that into Lambda as well so that it will be shter code but let's say we have this object here which is mapper and this is our Anonymous in our class so basically uh we have to do something here so what we have to do is this result set will have all the records right we have to take one record time so what map row

01:48:22
does is it takes the entire result set and it executes map R multiple times depend upon how many rows you have so if you have four rows the map row will execute four times and it will give you one object at a time so basically we'll take that we'll create object of alien here I will say alien a is equal to new alien and then this alien object is empty right as of now and I've written the same object so you can see row mapper needs object of alien so I'm getting the object here and then I'm

01:48:54
returning it but then this is an empty object right so it will not have values the values are there in result set so I can simply say a do set ID and now get the value from result set so you can say get int so how do you get data from result set you have to say get int and pass the column number so either you can say column number or you can say column name both work so here you can also mention ID a do set name this will be coming from the second row so get string I will say two here and then a do

01:49:25
set Tech RS do get string three as I mentioned before you can also use a number or you can use a name of the column and that's it once you got the data from result set you are giving it to the alien object and then you are returning it so basically R mapper will take care of the data one by one now the same object which is of mapper you have to pass it here so you're not just passing a query you are also specifying what to do with the result set so if you click on query it returns you the object

01:49:59
of list of aliens so the type is T alien so we have to accept the list of alien I will say aliens is equal to this so basically this query will return the list of aliens and that's what you are going to return not the new at least so what we did is we fired the query we got the data it's just that we have to map it we have to convert that data from result set into the list of aliens and that's what we have done with the help of mapper and that's it I hope this will work let's see let me remove some extra

01:50:31
spaces there so with this if I run this code and again I hope it will work let's run okay can you see that it worked so basically alien 101 name kidan technology blockchain 102 H AI 103 suil iot and then the new entry which we have added 111 naven and Java this is what we have added right so

basically these things are working out that's how you work with the spring boot with your jdbc and yeah that's it in fact you can do some more changes here if you want uh since this is a function interface we

01:51:05
can use Lambda expression here so we can remove from here to here cut we can also remove one cly brackets here and you have you just have to put the arrow now let's remove the SQL exception let's put the arrow and then in Lambda we don't even need the type so we can remove both the types here and you know one more thing the row number is not not getting used right so we can even use R there or any other variable if you want let's say a again it will not make it readable right because a is already there let's

01:51:36
say row number or we can say okay so this is the shorter code now you can also replace this mapper with this code again this is optional if you want to do it you can try to use Lambda not compulsory so we don't need to create the the reference here we can directly use this right so that's how we can do it in fact you can also remove this part you can directly write this particular statement in return even that works so if I run this code once again so you can see it works so yeah that's your spring

01:52:06
Boot jdbc and if you observed everything is happening with the help of jdbc template so you don't have to basically do all this configuration by yourself writing the connection statement creating the statement everything will be taken care by the template you just have to specify the query and fire the [Music] query welcome back everyone in this module we'll talk about spring MVC now when you create a spring MVC project there are two ways of doing it so in the earlier days we were having only one way

01:52:36
which was spring MVC configuration but now we also got spring boot so the idea of spring boot is just to make thing easy okay so in Spring MVC as well we can achieve all the thing but just that we have to do a lot of configuration and to solve that we have spring boot okay but what is spring MVC to understand spring MVC we have to understand what is MVC and to understand MVC we have to understand why we have MVC so let's go back to the old days when you talk about normal servit and JSP all let's talk

01:53:06
about servlet so whenever a client sends the request to the server it is server's responsibility to process the request and give it back to the client but as a client what a client will get so of course client need two things the first thing is client needs data of course right if you go to any website the first thing you need is data but you want that data in a well formatted way example when you go to Facebook on Facebook you see data plus you also see a proper layout of course every one will get the

01:53:37
same layout so if I go to Facebook and you go to Facebook we'll be seeing the same screen the same colors right what will change is data so your friend list different from my friend list your feeds will be different from my feeds what will remain common is design right so we thought as a s instead of creating everything because subet on the server side will create the layout and will create data yes you can do that but you know it becomes clumsy we feel that you know you should be have a separate task

01:54:06
or we we should be having separate things for separate task example we need someone who will accept the request we need someone who will give the design okay uh it's like a normal template so design will remain common what will changes data and to hold that data we want something so basically we want something which will accept the request we want something which will be viewed to the client and we want something where it will hold the data so we got a controller who will accept the request

01:54:35
who will send the response we want a view technology and then we want a model so model basically will hold the data and that's where we got this concept of MVC model view controller so if you have model view controller it will almost work same way as normal seret without using MVC but then the advantage would be we are separating the concerns okay we are separating the task so the controller will only accept the request in the earlier days we used to use salet for controllers and JSP for View and

01:55:06
normal poos or normal Java classes for model right it's it's that simple but we got some issues here the issues were you know whenever a client sends a request request goes to a controller right and we are assuming that you only have one One controller but what if you have multiple controllers oh that's an issue so let's say in application we have user controller we have accounts controller we have laptop controller we have so many controllers right as a website as a web service you will provide so many

01:55:33
services how do we create a multiple controllers very simple create multiple servlets one seret can handle only one request or one subet will be known as a single action controller okay so one server can handle only one request so we wanted something where you can have one class or One controller can handle multiple request that's one second we wanted something in front so we wanted something as a front controller what I'm saying is when you send a request instead of sending that request to the

01:56:03
actual controller first you will send the request to the front controller now based on what request you are sending it will check and then it will send it to the actual controller then now you will think okay we have to create a controller plus we have to also create front controller don't worry the the moment you start using framework like spring where you have spring MVC in Spring MVC you don't have to create front controller spring MV will give it to you of course the normal controllers

01:56:30
you have to build it uh so example normal controller is your user controller accounts controller laptop controller and this front controller will be your dispatch servet and dispatch servet will be given by Spring framework so now what will happen is the moment you send a request request goes to dispar outlet now let's say you want all the users you want to fetch all the users of course this can be done using user controller but who will call user controller and that's why disp says hey

01:56:57
don't worry I'm there so disp Ser will send your request to the user controller but disp is something which will given by Spring framework right so how would disire knows that you want to call user controller and that's where you have to create a file where you will configure all this things oh now

you have to also create the file it's not that difficult it's damn easy to work with but then again based on what you're using if you're using normal spring MVC then you have to do some sort of configuration if

01:57:24
you're using spring boot this configuration goes down we will see that but we have to configure it we have to configure your disp salet to call a particular controller and that's how spring MVC works so yes you can also use normal MVC for Serv JSP but if you're working on Spring using spring MVC will help you in your [Music] project so let's see the Practical implementation of spring MVC now of course when you talk about spring MVC as well we have two options here we can go with a normal spring MVC application or

01:57:58
we can go for spring boot MVC now of course there's advantage of using spring boot here you have to do less configuration so the way we did with spring core we'll start with spring boot now so we'll try to implement spring MVC with the help of spring Boot and then once you got the G of it we can go with the normal configuration yes normal configuration needs more setup but but then we can do that later so let me create a very simple spring boot application but this time I want to go

01:58:25
for spring MVC so let's right click here and say new and of course right when you say spring boot application I want to say spring starter project now this is where you have to mention group ID here we'll say delisco and have the artifact ID as spring MVC Boot and of course the package name I will say teliscope do spring MVC see boot click on next okay now what to select from here now since we are building a web application you have to say web now when you search for web you can see we have an option of web

01:59:01
here you can select this one you only need this one if you are going for reactive web you can just select this one which supports which uses web flux otherwise if you want to create a web service you can go for Jersey but as of now let's go with web and click on next we don't need anything else now we are not working with database or something let's click on next and and that's it click on finish the moment you click on finish it will give you a project so once you got your spring boot project

01:59:26
and you can see we got our project okay still getting downloaded you can see it takes some time to get it from the internet so let's wait for it okay so you can see we got our project here which is demo I guess I forgot to mention the name for the project we have assigned the artifact now to verify that we can go to pom.xml and here you can see we have the artifact ID as spring MVC boot okay that's great we forg for to change the name from demo to Spring boot MVC but that's fine you can see we

01:59:54
got one dependency which is very important here which is spring boot starter web the moment you add this dependency in your project this is your spring MVC now yes with the help of spring boot now first of all why do we create a web application of course there there will be a client and client will send you a request now based on the request you will process the request you will uh generate some data and you will send it back that's what we do right okay but in this scenario since we have not created anything which will accept

02:00:24
the request normally we call them as controllers uh so we have a concept of MVC where we have model view controller now it is a responsibility of the controller to accept the request from the client and then it will do some operation it will give the data back to the user normally we save that data in the model object and then we have a view which is actually your the layout which you want to send to the client now now that view can be JSP it can be template as well at the start at least we'll work

02:00:54
only on JSP and then we can see later okay so we got this demo here now what I want to do is I just want to create a controller which will accept the request let's say I want to call a homepage and I want to return a JSP now first of all we don't have a JSP you can see that I have not created any JSP here so let's open this SRC and here in this main I want one more folder so normally if you want to keep your JC Pages you have to keep that in a web app folder so right click here we'll send new I want to

02:01:25
create a folder now and we'll name this folder as web app okay now once you got your folder we want to create a JSP page let me do that quickly so I will right click here I will send new I want to create this uh I want to create a JSP a JSP file click on next okay now let's name this I will simply call it as index.jsp that what makes much more sense right and you can see in the web app we got our JSP p in this JSP I will say welcome to telescope that's it I just want to say welcome to telescope

02:01:55
and so whenever I call a homepage I want to just display this page but will this work okay let's we'll try that later now there's one thing you have to observe the first thing is uh we got JSP that's awesome that we have to put that in web app normally when you create a web application you always add the Tomcat server in your Eclipse right so in fact this is STS but you can imagine that is it is your eclipse and you can see in the server tab you can see we don't have any server here oh that means we have to

02:02:24
also do the setup okay we'll see that later but that means if you want to this application we need a tomet otherwise it will give you error what if I say you can simply run this application and it will not even ask you for the Tomcat let's try so when you right click on this application and when you say run as the only thing is you have to run it as a spring boot app not on server because if you say on on server it will ask you for the external server we don't have it let's click on Spring boot app the

02:02:51
moment you do that you can see it is doing something it will take some time so you can see if I jump to console it says spring boot app started it is doing some processing now okay now the amazing thing is it says Tomcat started on port number 8080 that's weird because we don't have any tomat server here right from where you're getting it the amazing thing is if you expand this me dependencies here spring boot application has a embedded Tomcat so you can see we have AED Tomcat here the advantage is the moment you

02:03:23
move your application to some other server some other machine you don't have to configure your own tomcat and that's the the benefit here so in the application itself you have your Tomcat server so whenever you carry your application you are also getting doget server that's great right okay so that's something amazing but how do we access this now of course it says you have to use the port

number which is 8080 so what I will do is I will open my browser so that's my browser I will simply

02:03:47
search for local host colon 8080 and the you say enter you can see we got something okay we got this error that's fine at least we are not getting that weird full page 404 error we got this 404 but then this is getting returned by the spring MVC framework it says there's no error page because we have not built it yet uh it shows the timing and there's no message that's it at least something is working we have not got the output it is not calling index.jsp but why it is not calling

02:04:16
index.jsp of course right when you send a request the request goes to the controller that's how spring MVC work so the first request goes to dispatcher serlet and from dispat server it will call a particular controller we don't have a controller yet how do we do that that we'll see in the next [Music] video so let's solve that issue the issue was we don't have a controller so let's create one now of course right when you say a controller we are working on Spring Frameworks it should be a Java

01:55:06
normal poos or normal Java classes for model right it's it's that simple but we got some issues here the issues were you know whenever a client sends a request request goes to a controller right and we are assuming that you only have one One controller but what if you have multiple controllers oh that's an issue so let's say in application we have user controller we have accounts controller we have laptop controller we have so many controllers right as a website as a web service you will provide so many

01:55:33
services how do we create a multiple controllers very simple create multiple servlets one seret can handle only one request or one subet will be known as a single action controller okay so one server can handle only one request so we wanted something where you can have one class or One controller can handle multiple request that's one second we wanted something in front so we wanted something as a front controller what I'm saying is when you send a request instead of sending that request to the

01:56:03
actual controller first you will send the request to the front controller now based on what request you are sending it will check and then it will send it to the actual controller then now you will think okay we have to create a controller plus we have to also create front controller don't worry the the moment you start using framework like spring where you have spring MVC in Spring MVC you don't have to create front controller spring MV will give it to you of course the normal controllers

01:56:30
you have to build it uh so example normal controller is your user controller accounts controller laptop controller and this front controller will be your dispatch servet and dispatch servet will be given by Spring framework so now what will happen is the moment you send a request request goes to dispar outlet now let's say you want all the users you want to fetch all the users of course this can be done using user controller but who will call user controller and that's why disp says hey

01:56:57
don't worry I'm there so disp Ser will send your request to the user controller but disp is something which will given by Spring framework right so how would disire knows that you want to call user controller and that's where you have to create a file where you will configure all this things oh now

you have to also create the file it's not that difficult it's damn easy to work with but then again based on what you're using if you're using normal spring MVC then you have to do some sort of configuration if

01:57:24
you're using spring boot this configuration goes down we will see that but we have to configure it we have to configure your disp salet to call a particular controller and that's how spring MVC works so yes you can also use normal MVC for Serv JSP but if you're working on Spring using spring MVC will help you in your [Music] project so let's see the Practical implementation of spring MVC now of course when you talk about spring MVC as well we have two options here we can go with a normal spring MVC application or

01:57:58
we can go for spring boot MVC now of course there's advantage of using spring boot here you have to do less configuration so the way we did with spring core we'll start with spring boot now so we'll try to implement spring MVC with the help of spring Boot and then once you got the G of it we can go with the normal configuration yes normal configuration needs more setup but but then we can do that later so let me create a very simple spring boot application but this time I want to go

01:58:25
for spring MVC so let's right click here and say new and of course right when you say spring boot application I want to say spring starter project now this is where you have to mention group ID here we'll say delisco and have the artifact ID as spring MVC Boot and of course the package name I will say teliscope do spring MVC see boot click on next okay now what to select from here now since we are building a web application you have to say web now when you search for web you can see we have an option of web

01:59:01
here you can select this one you only need this one if you are going for reactive web you can just select this one which supports which uses web flux otherwise if you want to create a web service you can go for Jersey but as of now let's go with web and click on next we don't need anything else now we are not working with database or something let's click on next and and that's it click on finish the moment you click on finish it will give you a project so once you got your spring boot project

01:59:26
and you can see we got our project okay still getting downloaded you can see it takes some time to get it from the internet so let's wait for it okay so you can see we got our project here which is demo I guess I forgot to mention the name for the project we have assigned the artifact now to verify that we can go to pom.xml and here you can see we have the artifact ID as spring MVC boot okay that's great we forg for to change the name from demo to Spring boot MVC but that's fine you can see we

01:59:54
got one dependency which is very important here which is spring boot starter web the moment you add this dependency in your project this is your spring MVC now yes with the help of spring boot now first of all why do we create a web application of course there there will be a client and client will send you a request now based on the request you will process the request you will uh generate some data and you will send it back that's what we do right okay but in this scenario since we have not created anything which will accept

02:00:24
the request normally we call them as controllers uh so we have a concept of MVC where we have model view controller now it is a responsibility of the controller to accept the request from the client and then it will do some operation it will give the data back to the user normally we save that data in the model object and then we have a view which is actually your the layout which you want to send to the client now now that view can be JSP it can be template as well at the start at least we'll work

02:00:54
only on JSP and then we can see later okay so we got this demo here now what I want to do is I just want to create a controller which will accept the request let's say I want to call a homepage and I want to return a JSP now first of all we don't have a JSP you can see that I have not created any JSP here so let's open this SRC and here in this main I want one more folder so normally if you want to keep your JC Pages you have to keep that in a web app folder so right click here we'll send new I want to
....give important and teach me like this only in very very simple

Samiksha…
YES! I will teach you **JUST like this — same detailed story style, same slow flow, same "step-by-step with brain connection" style**, BUT **in super simple language** so you can understand everything clearly.

You pasted a *huge* explanation (Spring Core → Setter Injection, Constructor Injection, Autowire, Primary… then Spring Boot JDBC → Table creation, RowMapper, JDBC Template…).
You want **important concepts** explained like the video, but in **your learning style — extremely simple, real-world, slow, with metaphors**.

So I'll now teach you **every important concept** from your text in a simple, practical, beginner-friendly way.

✅ PART 1 — Spring Core (Beans, DI, Setter, Constructor, Autowire, Primary)

⭐ 1. What is a Bean?

Bean = Object created by Spring container.

Real life:
Think like a **kitchen**.
Spring is the **chef**.
You only tell the chef the recipe (class), chef prepares the dish (object).
You don't cook it manually.

```java
<bean id="alien" class="com.Alien"/>
```

⭐ 2. Setter Injection

When Spring sets values using the **setter method**.

Example:
Alien has:

```java
int age;
Laptop laptop;
```

In XML:

```xml
<property name="age" value="23"/>
<property name="laptop" ref="laptop"/>
```

Simple rule:

- **value** → **primitives** (int, string)
- **ref** → **object references**

⭐ 3. Reference Injection

If Alien has a Laptop object:

```java
Laptop laptop;
```

Then in XML:

```xml
<property name="laptop" ref="laptop"/>
```

Ref means: “Spring, give me the object of bean id = laptop.”

⭐ 4. Constructor Injection

If Alien has a constructor:

```java
Alien(int age)
```

Then in XML:

```xml
<constructor-arg value="25"/>
```

Setter vs Constructor — When to use what?

Case	Use
Value is **mandatory**	**Constructor Injection**
Value is **optional**	**Setter Injection**

Because constructor forces the value, setter doesn't.

⭐ 5. Autowire

Autowire = Spring will **automatically find and inject** the dependency.

Two types:

✔️ By Name

Matches the property name with bean id.

```xml
<bean id="com" class="Laptop" autowire="byName"/>
```

Alien has:

```java
Computer com;
```

Spring sees `com` property and `com` bean → connects automatically.

✔️ By Type

Matches **class type**.

```xml
<bean id="desk" class="Desktop" autowire="byType"/>
```

Alien has:

```java
Computer com;
```

Both Desktop & Laptop implement Computer → confusion!

Spring says:

> "Two beans found for type Computer. I'm confused."

⭐ 6. Primary Bean

To tell Spring which bean to prefer when multiple match exist.

```xml
<bean id="laptop" class="Laptop" primary="true"/>
```

Laptop becomes preferred.

✅ PART 2 — Spring Boot JDBC (Super Simple)

⭐ 1. Why JDBC Template?

Normal JDBC = too many steps →
Driver load, Connection, PreparedStatement, ResultSet, Close...

JDBC Template = **Spring does everything**, you write only query.

⭐ 2. Components you created:

You made:

✔️ Model → Alien

Represents table columns.

```java
int id;
String name;
String tech;
```

This class = one row from the table.

✔️ Repository → AlienRepo

Responsible for database connection.

Marked with:

```java
@Repository
```

Spring will manage it.

⭐ 3. Saving data using JDBC Template

```java
String sql =
"INSERT INTO alien (id,name,tech) VALUES (?,?,?)";

template.update(
    sql,
    alien.getId(),
    alien.getName(),
    alien.getTech()
);
```

update() is used for insert/update/delete.

⭐ 4. Error: “Table not found”

Because H2 starts empty.

Solution → create:

```pgsql
schema.sql   (table structure)
data.sql     (initial data)
```

Spring auto-runs these files at startup.

Example:

✔️ schema.sql

```sql
create table alien (
  id int primary key,
  name varchar(50),
  tech varchar(20)
);
```

✔️ data.sql

```sql
insert into alien(id,name,tech) values
(101, 'Kiran', 'Blockchain'),
(102, 'Harsh', 'AI'),
(103, 'Sahil', 'IoT');
```

⭐ 5. Fetching data using query()

```
java

String sql = "select * from alien";
return template.query(sql, new RowMapper<Alien>() {
    public Alien mapRow(ResultSet rs, int rowNum) {
        Alien a = new Alien();
        a.setId(rs.getInt(1));
        a.setName(rs.getString(2));
        a.setTech(rs.getString(3));
        return a;
    }
});
```

RowMapper converts each row → Alien object
query() returns List<Alien>

⭐ PART 3 — Spring MVC (Super Simple)

Why MVC?

To separate responsibilities:

Part	Responsibility
Controller	Accept request, call service
Model	Holds data
View	JSP/HTML page shown to user

Front Controller

Spring gives a built-in **DispatcherServlet**.

Flow:

```
arduino

Client → DispatcherServlet → Controller → Model → View (JSP)
```

Basic Controller Example

```
java

@Controller
public class HomeController {

    @RequestMapping("/")
    public String home() {
        return "index.jsp";
    }
}
```

⭐ Samiksha, do you want me to continue like this for:

✅ Full Spring MVC
✅ Service Layer
✅ Watching step-by-step example (Add, Update, Delete with MVC)
✅ Spring Boot + JPA (VERY easy compared to JDBC Template)
✅ REST APIs
✅ Project-level explanation?

Just say **"Yes continue in this style"** or tell me which part next.

assign a default age because if you don't if you don't Define the default age by default will be zero I don't want 1 Zer at least the age should be 12 or maybe 10 how do I mention that 10 here of course there's one way you can just go here and say 10 right by default the age value will be 10 that's one option otherwise you can just go to app when you create the object you can assign it manually you can pass it to the Constructor what if you want to assign the value in the spring Bean itself

01:01:17
because remember the object will be created by Spring framework so the Constructor will be called by Spring framework right so I want to assign the value for age here in this XML file can we do that let's try so let me just expand this bin now in this be tag we can have one more tag here which is named as property now what is property so every variable in your class if your class is been every variable will be known as property you can have multiple properties so that's the property here

01:01:47
you have to mention the name of your property in this case it is age so we have name and now we have to also specify value so to specify the the name and the value if your property is a primitive value or primitive type you can assign value but what if your property is of type reference that we'll see later but time let's only focus on value I want to assign the value as 10 by default the age should be 10 that's what we want here but will this work let's try go back to app and you can see

01:02:15
I have not assigned the value for age anywhere else apart from that your XML file let's run this code and you can see we got 10 so what happens here the moment you run this application on this line you will get that particular container in your jvm in that container it will check hey we have a springl file where you have mentioned about alien object so it is my job to create that alien object the moment you create the object it will also call the set age method you know why because we are saying property since you are specifying

01:02:48
property it will call set age oh okay I know you don't trust me here let's go back to alien and oh we don't have a a Constructor as well we have a Constructor here so let me just put this Constructor on top so that it will be visible right I also want to go to set age and by saying here age assign that's it just to see if that set me is getting called let's go back to app and now let's relaunch the application and you can see it says alien object Creator that's step one

01:03:19
second step is age assigned so first it will create the object second and it will set the age because we are saying property right just to prove my point once more what I will do is I will go back here and let me just change the method name from set age to set age one now what will happen is if I do that if I go back to app and let's relaunch the application you can see we got an error it's compulsory to have your method name suitable for your variable so if your variable name is age it should be age

01:03:49
right it should not be one that's what it's it's saying here you can see it says Ed creating be with name Alien because if you go you know if you go there it says invalid property age because we don't have it there the property we are using here is set age one right make sure that you do set age yeah if you want to keep it set age one in XML as well you have to say age one then it will work right okay so let's go back and you can see if I say age it will it is giving me error so let's go

01:04:18
back to our code and it set age so that that's why this set methods and get methods are so important in Spring so that's about your how do you assign a default value but what if you have a reference type now when you say Age It's A Primitive right a int but what if a alien needs a laptop now of course laptop will be a class then the laptop will have certain methods and then we have to create the reference of laptop in this alien okay when you say reference how will you assign the particular value to it because we cannot

01:30:52
you're using any springing orm technology let's say hibernate in that case the table creation happens automatically now since we're not using hibernate here we have to create the table by ourself yeah that's something you have to do okay let's create a class here and I want to create a class for the let's say alien okay so when I say alien class what I want to do is I want to basically create few variables so I will say private in ID and then private string name and private string

01:31:22
Tech so basically I need these three variables and for these three variables I also need Getters and Setters so we'll right click here and we'll say generate Getters and Setters I will just choose these three variables done I also need a two string method if I want to print the data itself so I will just right click here say generate uh two string method and done I will do one more thing I will move this particular class the alien class to a different package so this should be into a a package of model so every class

01:31:54
should belong to their own packages so alien is a model class so I will just move that to a model package and since we don't have a model package I will say hey move to model package in the SRC yes and you can see it is moved now so that's done but if I want my spring framework to manage this particular bean I have to use add component now if you're not sure how spring boot Works uh you you can check out the video link in the description I will make sure that you have your spring boot video there

01:32:21
okay so basic basically what we are doing with ADD component is we are asking spring framework to handle this object creation object managing and object destroying by Spring framework I don't want to do that okay so basically when I say I don't have to create the object how spring will do it

so basically what I will do is uh this run method so I will go back to the main class so main class is your spring jdbc demo application and in this the you can see we are calling a run method run Returns the object of application

01:32:48
context so I will say application context cont text now why I'm doing this is because I want to get the hold on a spring container again I'm assuming that you know spring boot here because if I start teaching your spring board then it will take it will make the video length of five hours I don't want to do that so again you'll find the link in description you can check it out now let me create the object for alien so I will say alien I will say alien one is equal to okay we have to basically import the

01:33:12
package for alien which is in the model package done and now I want to create the object the way you can create the object by saying alien right but I don't want to create this object by myself I just want this to be done by Spring framework so I will say context. get bean I want the object for alien do class right uh but the thing is by default the type of this alien component is Singleton so basically the ban is Singleton I want to make sure this is prototype so I will say scope prototype

01:33:41
so what will happen is for different request it will create new objects for you otherwise you will have multiple alien object or multiple alien references with the same object so you will have the same data doesn't matter how many data you have in your table okay so that's done and for this alien what I will do is I will add some values so let's say the ID I want to add is 111 and I want to say not get ID I want to say set ID alien do set name I want to set the name as let's say naven and

01:24:22
that but the only problem is the more code you write you are basically increasing the number of lines which will have a bug so you know every new line is a potential bug right so reduce the number of lines which you write and use certain tools of Frameworks which will do it for you and these Frameworks and tools are been tested by everyone so if it works it works for you as well now that's where we have to use a template now there's something called jdbc template now jdbc template will help you

01:24:50
with data source that means the connection pulling concept will be done and it will also help you to write your code in an efficient way and all these things are provided in Spring boot jdbc now why spring boot is because if you want to build Enterprise application and if you want to follow all the design patterns best practices and a framework where you can get the libraries and everything working together spring is one of the best framework and that's why we're going to use spring boot jdbc is

01:25:15
because under the spring umbrella we have lot of projects and this is one of it so let's get started with the code where we going talk about how do you build an application and we'll learn by building something right so it will make much more sense so let's start the code the thing is if you want to build this project you need a IDE now you can choose any ID which you want maybe you can use Eclipse intellig vs code your choice I'm going to use intell idea and that to a community version because

01:25:44

that's for free and it works so we'll be using that so to get the project for spring if you're using a spring tools on vs code or Eclipse you can directly get the project from there and if you're using the ultimate version of intell idea you can create the project directly from your IDE but since we are using a community version I'm going to use the spring inst viser so you can head to the start. spring.io and you can get your project from here the base skeleton of the project so the first thing you will

01:26:15
mention is a project type I'm going to go with mavin uh language I will choose Java of course that's what you're doing now and then the spring boot version I'm selecting 3.2 next you have to mention the group ID so I'm going to select I'm going to type here com. telescope the project since I'm working with spring jdbc demo that's the project name I I will give and will not change anything here the packaging I want it to be type of jar and the version for Java I'm

01:26:44
using in my machine is 21 now based on which version you have you can either just choose 21 or you can select 17 uh dependencies so basically we have to add uh two things only one is the jdbc API now this is your springboard jdbc and you have to add a database now which database we going to use of course you can choose any dbms which you want now we will be using a embedded one so if you have post msql Oracle mssql you can use that the only thing is you have to do some additional configuration and

01:27:18
I will tell you where to do that configuration but we are going to use a inmemory database here and by default when you use inmemory like hed2 it will be pre-configured so you don't have to do any configuration for that so this these are the two libraries we need and that's it you can click on generate it will give you a project so as you can see I got the project downloaded which is spring jdbc demo I will just unzip it and open in any ID which you want I'm going to use intj so let me do that

01:27:46
quickly okay so basically have opened that project in my IDE and it will take some time for it to resolve the dependencies now we have added two which is jdbc and the hed2 and we don't have to manually download the H2 driver because the moment you add H2 it will also have a driver installed so if I open the pal. XML you can see we do have H2 here and we got spring boot starter jdpc this is what we want and now we can get the work started so we'll go back here to main Java and spring jdbc demo

01:28:20
application and we can start the coding here okay so one thing I mentioned before right if you want to use P or Mas or any other dbms variable to the configuration so if you can see we have a resources folder here and if you open application. properties this is where you have to specify those properties so what properties I'm talking about you have to specify the URL you have to specify the username and password and Driver now basically this variable belongs to spring. dat source. URL and you can set the value for the URL

01:28:51
whichever you want so that's how you do it but now we are not going to use Mas or pus we are going to use H2 which is in memory reason I'm doing using H2 here is because you can directly take this project and run on your machine if I use P or MySQL then you have to also install those softwares you have to configure it and there will be some issues with that so let's go stick to H2

okay so what we want to do we want to basically store some data in the database and fetch it from database so we do have database

01:29:20
ready so the moment you run this application you will be having Edge to database behind the scene running for you but then what you want to store so in that case what I will do is I will create a simple class which will represents a particular table so normally you can have a relation between a table and your class now you will say how is is it possible see the thing is if you think about a table table will have a table name table will have a columns let's say we are working with aliens so every

01:29:48
02:01:25
create a folder now and we'll name this folder as web app okay now once you got your folder we want to create a JSP page let me do that quickly so I will right click here I will send new I want to create this uh I want to create a JSP a JSP file click on next okay now let's name this I will simply call it as index.jsp that what makes much more sense right and you can see in the web app we got our JSP p in this JSP I will say welcome to telescope that's it I just want to say welcome to telescope

02:01:55
and so whenever I call a homepage I want to just display this page but will this work okay let's we'll try that later now there's one thing you have to observe the first thing is uh we got JSP that's awesome that we have to put that in web app normally when you create a web application you always add the Tomcat server in your Eclipse right so in fact this is STS but you can imagine that is it is your eclipse and you can see in the server tab you can see we don't have any server here oh that means we have to

02:02:24
also do the setup okay we'll see that later but that means if you want to this application we need a tomet otherwise it will give you error what if I say you can simply run this application and it will not even ask you for the Tomcat let's try so when you right click on this application and when you say run as the only thing is you have to run it as a spring boot app not on server because if you say on on server it will ask you for the external server we don't have it let's click on Spring boot app the

02:02:51
moment you do that you can see it is doing something it will take some time so you can see if I jump to console it says spring boot app started it is doing some processing now okay now the amazing thing is it says Tomcat started on port number 8080 that's weird because we don't have any tomat server here right from where you're getting it the amazing thing is if you expand this me dependencies here spring boot application has a embedded Tomcat so you can see we have AED Tomcat here the advantage is the moment you

02:03:23
move your application to some other server some other machine you don't have to configure your own tomcat and that's the the benefit here so in the application itself you have your Tomcat server so whenever you carry your application you are also getting doget server that's great right okay so that's something amazing but how do we access this now of course it says you have to use the port number which is 8080 so what I will do is I will open my browser so that's my browser I will simply

02:03:47
search for local host colon 8080 and the you say enter you can see we got something okay we got this error that's fine at least we are not getting that weird full page 404 error we got this 404 but

then this is getting returned by the spring MVC framework it says there's no error page because we have not built it yet uh it shows the timing and there's no message that's it at least something is working we have not got the output it is not calling index.jsp but why it is not calling

02:04:16
index.jsp of course right when you send a request the request goes to the controller that's how spring MVC work so the first request goes to dispatcher serlet and from dispat server it will call a particular controller we don't have a controller yet how do we do that that we'll see in the next [Music] video so let's solve that issue the issue was we don't have a controller so let's create one now of course right when you say a controller we are working on Spring Frameworks it should be a Java02:04:46
file let's do that so let's go back here and here I want to create a controller oh how do we do that let's right click here and say new I want to create Java class let's select that and this will be of course you can name anything right from ABCD to some logical name but here I will say this is my home controller so logically when you say you have an application your application will have multiple Services let's say you have a Accounting Service uh let's say you have a user service you have uh let's say

02:05:17
some some other service so when say you have a user service and you have certain request which a server will handle what makes most sense is to have a controller named as user controller if you're working for accounting so it should be accounting controller now since we are working with homepage let's say home controller and that's it nothing to specify much here and click on finish and that's your controller okay now you will say hey this will not work how can you just simply create a controller and

02:05:46
you're getting the I mean the normal class is a controller kind of you don't have to extend or Implement any interface here the only thing you have to add to this controller is a annotation which is ADD controller this is what makes it as a controller now as we discussed your request will first go to dispatch a serlet and despite serlet will send it to a particular controller okay now yes in theory we have dispite serlet needs a configuration file as of now you can see we have not created any

02:06:16
configuration file and that's why spring boot says hey uh you don't have to worry about the default configuration I will take care of it yes if you want to provide some extra configuration that's where you have to use some configuration file as of now it will work so we got a controller now this controller whenever you call this controller you have to perform certain operation oh when you say operation that means we need a method let's create one so I will say public and this method name will be

02:06:42
let's say home of course uh you can have a different method name but that's fine let's give it home I want to call this method home which will simply print as of now I'm not calling any JSP let me just check if this meth is getting called when I call the homepage uh I will say homepage requested that's it nothing much I'm just printing homepage requested okay but will this work let's restart the application let's say restart the application it will take some time so let's keep our page ready

02:07:11
and let's refresh and you can see after refreshing the only thing we changed is the timing okay we

have not got the output yet but of course the output you be getting is in the console right because you're saying system.out.println and if you can see the console we have no way it says homepage requested that means is not working the reason it is not working is you know you might be having different kind of request you're requesting for homepage you might be requesting for some particular Page by

02:07:35
saying slash accounts or slash users or maybe slash laptops here I want to specify Hey whenever I use a request for the homepage then only you have to call this method and the way you specify that is with the help of of annotation here which is request mapping so after say request mapping in bracket you have to mention the path first of all you have to import the package let's say control space package imported and here you have to say slash so that's our homepage right when you say the slash okay I

02:08:04
guess it will work now let's relaunch the application okay let's refresh it and you can see oh we got a different type of error this time we got internal server ER we got 500 okay we'll talk about that later what is happening but let's see the console here are we getting homepage requested and the answer is yes can you see that we got the output it says homepage requested that's what we want it right yes we got some error we'll talk about that error in some time but it says homepage

02:08:31
requested so yes we got a controller we got a JSP the only thing is this controller is not able to call the GSP page that's what we have to solve now now how do we solve it when you send a request for the home okay if you check your url it says lo host colon 8080 and then we are saying slash okay that's how you request the homepage so this we is getting called so we have to inform here whenever you call home method we have to call index.jsp now if you're coming from Ser background we used to uh create a

02:09:03
request dispatcher using which you used to call the GSP here we don't have to do all those thing we simply have to return the name of the JSP which you want to call so you have to say index.jsp so what happens is see who is calling this home controller it is called by dispatcher servlet so if dispatcher Ser is calling this method so it is the job of dispat serlet to call the actual page so we just have to inform hey which page you want to call so we are saying hey dispat Ser I want to call index.jsp that's it the only

02:09:38
thing is we have to say string here and our job is done let's relaunch it so as you can see it is relaunched let me just open my browser and say refresh oh okay something happened you can see okay I saw some animation there something happened uh we got the same error but if I expand this if you can see the downloads I just got a download here let me just relaunch it again and you can see we got another download what is happening the thing is okay so let me just go back again and let me just click

02:10:08
on this I don't know what this file is it is says unknown DMS just open that in a finder so as you can see if I open this it is my JSP page oh that's weird I'm downloading the JSP page what is happening so let's try to solve this problem in the next video so the problem is we are not able to run the GSP we are downloading it let's solve this issue in the next [Music] video so what's wrong here everything is working except one thing which is instead of running JSP You are

02:10:40
downloading it we don't want that we want to run that JSP on Tomcat so what happens you know when you talk about JSP JSP get converted into s and that sulet runs on Tomcat here nothing is happening we don't have something which will convert your JSP into seret now if you talk about the external Tomcat it has that power the Tomcat which we are using it normally don't convert JSP into seret okay in fact the spring boot is famous for rest Services you don't actually create websites in Spring boot that was not the

02:11:12
main idea the main idea was to build rest Services if you want to achieve that if you want to use GSP yes we can use them Le that perfectly works but if you want to use JSP and if you want to convert JSP into salet we need a special Library which you have to download and that is Tomcat Jasper so let's do that so let's go to our browser and let's go to the favorite website of us which is MAV repository and here we have to search for Tomcat Jasper so this is what you need so I will just download this

02:11:42
one okay now which version of Tomcat Jasper we need so let's go back to our application and here let's open open the m dependencies so you can see we have the Tomcat version it's 9.0.4 so let's stick to that so let's download the 9.0.4 uh let's copy this dependency I guess it is copied let's go back where you will add it of course you have to add that in a palm. XML file so open that and here let me add that particular dependency the moment you say paste and save it will download Tomcat Jasper from

02:12:14
the internet it will take some time now since I don't have the local copy of it so you will get the repository from the official website so as you can see we got dependency here we got Tomcat Jasper now once you have added that let's see let's run this application once more and let's see the output so let's relaunch the application okay let's go to browser and refresh and now you can see we got the output so this time we are not downloading it but we are running the JSP so that's the thing if you want to

02:12:39
run JSP you need a tomcat Jasper library to convert that so it is working we got welcome Tesco our job is done so whatever we have done till now is very simple we I have created a page which is index.jsp which is actually our web page and then we have this home controller where you are it's a controller basically and you can have a method which will accept the request for Slash and then it will simply call index.jsp but what if you want to do something extra let's say I want to have a page where user will enter some values

02:13:09
so let's say I want to say I want to add two numbers yes we can do something big as well but let's stick to simple because we are working on we are learning spring framework right we are not just learning logic here so let's keep the logic simple and let's focus on the on the concept so in the next video we'll talk about how do we add two numbers using spring [Music] MVC so let's get the application so of course if you want to add two numbers after calling the web page it will

02:13:38
instead of printing welcome to teliscope will also print the form where a user will enter two values so let's do that quickly so here I will say create a form for me and the form will have uh the the

action and I will say the action is ADD so I want to call Simply add and then form ends and here I want to create two text field so I will say okay so that's our first field and we want one more so I will just copy this and paste and now you can see we got the second one as well so we simply say second number and

02:14:07
here we'll have it as num two now what next I want to also have a button here right which will when you click on that it will call add uh let's do that so I will say okay that's done so we got our two put text for Num one num two and we also got buttons everything is done yes we don't have a perfect UI here but doesn't matter right we are learning spring uh so UI doesn't matter at this point yes it matters when you give it to your client as of now it should work now once you got that page of course when

02:04:46
file let's do that so let's go back here and here I want to create a controller oh how do we do that let's right click here and say new I want to create Java class let's select that and this will be of course you can name anything right from ABCD to some logical name but here I will say this is my home controller so logically when you say you have an application your application will have multiple Services let's say you have a Accounting Service uh let's say you have a user service you have uh let's say

02:05:17
some some other service so when say you have a user service and you have certain request which a server will handle what makes most sense is to have a controller named as user controller if you're working for accounting so it should be accounting controller now since we are working with homepage let's say home controller and that's it nothing to specify much here and click on finish and that's your controller okay now you will say hey this will not work how can you just simply create a controller and

02:05:46
you're getting the I mean the normal class is a controller kind of you don't have to extend or Implement any interface here the only thing you have to add to this controller is a annotation which is ADD controller this is what makes it as a controller now as we discussed your request will first go to dispatch a serlet and despite serlet will send it to a particular controller okay now yes in theory we have dispite serlet needs a configuration file as of now you can see we have not created any

02:06:16
configuration file and that's why spring boot says hey uh you don't have to worry about the default configuration I will take care of it yes if you want to provide some extra configuration that's where you have to use some configuration file as of now it will work so we got a controller now this controller whenever you call this controller you have to perform certain operation oh when you say operation that means we need a method let's create one so I will say public and this method name will be

02:06:42
let's say home of course uh you can have a different method name but that's fine let's give it home I want to call this method home which will simply print as of now I'm not calling any JSP let me just check if this meth is getting called when I call the homepage uh I will say homepage requested that's it nothing much I'm just printing homepage requested okay but will this work let's restart the application let's say restart the application it will take some time so let's keep our page ready

02:07:11
and let's refresh and you can see after refreshing the only thing we changed is the timing okay we have not got the output yet but of course the output you be getting is in the console right because you're saying system.out.println and if you can see the console we have no way it says homepage requested that means is not working the reason it is not working is you know you might be having different kind of request you're requesting for homepage you might be requesting for some particular Page by

02:07:35
saying slash accounts or slash users or maybe slash laptops here I want to specify Hey whenever I use a request for the homepage then only you have to call this method and the way you specify that is with the help of of annotation here which is request mapping so after say request mapping in bracket you have to mention the path first of all you have to import the package let's say control space package imported and here you have to say slash so that's our homepage right when you say the slash okay I

02:08:04
guess it will work now let's relaunch the application okay let's refresh it and you can see oh we got a different type of error this time we got internal server ER we got 500 okay we'll talk about that later what is happening but let's see the console here are we getting homepage requested and the answer is yes can you see that we got the output it says homepage requested that's what we want it right yes we got some error we'll talk about that error in some time but it says homepage

02:08:31
requested so yes we got a controller we got a JSP the only thing is this controller is not able to call the GSP page that's what we have to solve now now how do we solve it when you send a request for the home okay if you check your url it says lo host colon 8080 and then we are saying slash okay that's how you request the homepage so this we is getting called so we have to inform here whenever you call home method we have to call index.jsp now if you're coming from Ser background we used to uh create a

02:09:03
request dispatcher using which you used to call the GSP here we don't have to do all those thing we simply have to return the name of the JSP which you want to call so you have to say index.jsp so what happens is see who is calling this home controller it is called by dispatcher servlet so if dispatcher Ser is calling this method so it is the job of dispat serlet to call the actual page so we just have to inform hey which page you want to call so we are saying hey dispat Ser I want to call index.jsp that's it the only

02:09:38
thing is we have to say string here and our job is done let's relaunch it so as you can see it is relaunched let me just open my browser and say refresh oh okay something happened you can see okay I saw some animation there something happened uh we got the same error but if I expand this if you can see the downloads I just got a download here let me just relaunch it again and you can see we got another download what is happening the thing is okay so let me just go back again and let me just click

02:10:08
on this I don't know what this file is it is says unknown DMS just open that in a finder so as you can see if I open this it is my JSP page oh that's weird I'm downloading the JSP page what is happening

so let's try to solve this problem in the next video so the problem is we are not able to run the GSP we are downloading it let's solve this issue in the next [Music] video so what's wrong here everything is working except one thing which is instead of running JSP You are

02:10:40
downloading it we don't want that we want to run that JSP on Tomcat so what happens you know when you talk about JSP JSP get converted into s and that sulet runs on Tomcat here nothing is happening we don't have something which will convert your JSP into seret now if you talk about the external Tomcat it has that power the Tomcat which we are using it normally don't convert JSP into seret okay in fact the spring boot is famous for rest Services you don't actually create websites in Spring boot that was not the

02:11:12
main idea the main idea was to build rest Services if you want to achieve that if you want to use GSP yes we can use them Le that perfectly works but if you want to use JSP and if you want to convert JSP into salet we need a special Library which you have to download and that is Tomcat Jasper so let's do that so let's go to our browser and let's go to the favorite website of us which is MAV repository and here we have to search for Tomcat Jasper so this is what you need so I will just download this

02:11:42
one okay now which version of Tomcat Jasper we need so let's go back to our application and here let's open open the m dependencies so you can see we have the Tomcat version it's 9.0.4 so let's stick to that so let's download the 9.0.4 uh let's copy this dependency I guess it is copied let's go back where you will add it of course you have to add that in a palm. XML file so open that and here let me add that particular dependency the moment you say paste and save it will download Tomcat Jasper from

02:12:14
the internet it will take some time now since I don't have the local copy of it so you will get the repository from the official website so as you can see we got dependency here we got Tomcat Jasper now once you have added that let's see let's run this application once more and let's see the output so let's relaunch the application okay let's go to browser and refresh and now you can see we got the output so this time we are not downloading it but we are running the JSP so that's the thing if you want to

02:12:39
run JSP you need a tomcat Jasper library to convert that so it is working we got welcome Tesco our job is done so whatever we have done till now is very simple we I have created a page which is index.jsp which is actually our web page and then we have this home controller where you are it's a controller basically and you can have a method which will accept the request for Slash and then it will simply call index.jsp but what if you want to do something extra let's say I want to have a page where user will enter some values

02:13:09
so let's say I want to say I want to add two numbers yes we can do something big as well but let's stick to simple because we are working on we are learning spring framework right we are not just learning logic here so let's keep the logic simple and let's focus on the on the concept so in the next video we'll talk about how do we add two numbers using spring [Music] MVC so let's get the application so of course if you want to add two numbers after calling the web page it will

02:13:38
instead of printing welcome to teliscope will also print the form where a user will enter two values so let's do that quickly so here I will say create a form for me and the form will have uh the the action and I will say the action is ADD so I want to call Simply add and then form ends and here I want to create two text field so I will say okay so that's our first field and we want one more so I will just copy this and paste and now you can see we got the second one as well so we simply say second number and

02:14:07
here we'll have it as num two now what next I want to also have a button here right which will when you click on that it will call add uh let's do that so I will say okay that's done so we got our two put text for Num one num two and we also got buttons everything is done yes we don't have a perfect UI here but doesn't matter right we are learning spring uh so UI doesn't matter at this point yes it matters when you give it to your client as of now it should work now once you got that page of course when

02:14:33
you click on this button submit it will call add but don't you think we don't have anything here which will accept the request for add now think about this what can be done of course you need a page where you will show the output but before showing the output I want to process that information information so let's say if a user says 2 + 3 I want to give it five as answer where we do that so of course you need to create a controller so if you're coming from Ser background so what we do is we create

02:14:58
controllers but in Spring MVC you don't have to create a seate class every time what you can do is you can just add another method yes you can create a new class named it as calculator controller in which you can have different methods but as of now let's keep it here so in the same controller which is home controller we can have one more method so we'll do that I will say public this will return a string as of now let's keep it string and here I will say the method name as ADD of course you can

02:15:25
have any method name you you don't have to stick with ADD method name here and simply return the page you want to call of course right if you add these two numbers I want so on some page in fact instead of showing it on a page you can also return this as a data but we'll see that later how do we return a data as it is I want to call a page and we don't have a page as of now let's create one uh so we will just simply say copy and paste and I will name this page as result. JSP and in this JSP I don't want

02:15:53
the form and stuff I just want to print uh the output okay now how do we print the output so as of now simply thinging result is I'm not printing the actual output because we don't know the actual output let's go back here and in home controller you will simply say result. JSP that's it nothing much we have created two pages one is index.jsp and result. JSP okay so this looks fun okay I'm not sure if this is fun let me just try oh we're still missing something how do you know that you want

02:16:21
to call this add method for the add request of course it will not match the name you have to use a request mapping as The annotation So when you say request mapping in this you have to say add so whenever you call add request it will call add method is that simple let's run this so I will just

restart the application once so let's go back to our code and let's refresh once again so when you ref now you can see not just we got welcome with teliscope but we also got two text field here let's say two

02:16:47
and three three as of now we have not done any coding for that let's click on submit okay you can see it is actually calling result page okay not in the address bar because if you see the address bar we have still have uh num 1 equal to two and num 2 equal to 3 okay uh we are not specifying which way is getting called here but result. JSP is getting called because we got the output as result is okay that looks cool but uh we want to perform the operation right so here I want to add those two numbers

02:17:15
now how will you do it now let's go back to our salet and GSP stuff uh if you have learned it before so in salet what we do is if you want to take the input from the user uh we use this very special interface which is known as sttp solet request so using that you can fetch values so what I will do I will just go to this ad and I will say hey I want to use those methods if remember it's not just one parameter we us to pass two parameters so in this service method we us to pass the request object

02:17:45
and the response object we don't have anything here so we have to create them right so let's do that so I will say HTTP Sate request and you can say req and we got that object so we'll say we we got this object now you'll be thinking how spring will know that you want the instance of this interface here that's the magic of spring dependencies injection right if you remember we have talked about spring ioc or spring dependency injection where it will inject the object which you need now

02:18:12
thankfully this object is available with Tom cat so spring will say okay we have that object in Tom let's use it that's great now once you got that object we can fetch the values now so I will say int I where you will have the first variable value I know it's not a logical name to use but that's fine so we'll simply say request. get parameter you have to also convert this into integer right so you have to say integer. parcent Okay that's how you do it in fact for second variable as well we do the same thing

02:18:40
let's copy this and paste and this time we'll name it as J of course and here I will say num two now once you got those two values you can simply add them right how do we add it uh it's very simple you can say int I will have the variable name as maybe num three I don't know if it makes sense I will say I + J so the value of I and J will be getting added to give you num three now I want to print this num three because this is my output in the result page oh how would I do that so we have to send this num

02:19:11
three to the result page and to do that it's very simple actually you can simply say Hey how do you do that okay so in solet we used to use a concept of HTP session uh so when we use HTTP session we can simply add this num three in that object and in the JSP you can fetch it that's one way you can also use send disp where you can pass this value here you can say result. JSP question mark num3 equal to and whatever value of num three is you can say num three here that's URL rewriting right that's one

02:19:45
thing you can do we have multiple options right as of now what I will do I will simply say uh I will add it in a session so I will say HTTP session session is equal to Let's create the object of it oh but how do we do that so we okay first of all we have to say we have to say control space let's import the package and here I will say request dot get session so we have this method which is get session which will give you object of session okay once you have that you can simply say session dot you

02:20:15
can add not you can okay what's the method name I forgot the name you have to say set attribute and the attribute name we can have num three name can be anything right you can change it the only thing is whatever name you use here you have to use same name in the GSP page oh okay so will say num three in this case Okay once you got this attribute num three we are call result. JSP in result. JSP I just have to say uh we can use Java code here we can use session object and we can fetch the value or we can use jstl so as for the

02:20:46
jstl syntax we simply have have to say dollar and the cly packets and here you will say num three that's it so whatever name you have in the salet or in the Java code need to have it here okay it makes sense uh but but sometime sometime your gstl may not work so maybe your code is ignoring it so we will simply say is e ignored false or sometime it ignores e language which is expression language okay uh let's keep it just to be on safe side I'm saying false everything looks cool I think it will

02:21:19
work let's restart the application so let's go back to the page uh in fact I want to call the first page let's refresh in fact you can see the the address is Local Host colon 880 and then here I will say num one as three and num two as four or five uh so when you click on submit you want output as eight and we got it can you see that we got the output as eight so what is happening behind the scene uh so when you send a request the request is getting called for the homepage right so when you say

02:21:45
homepage this method is getting executed which is calling index.jsp and that's why we got that page which we have where we have welcome to telescope and then we have a form when you click on submit it is calling add request is again going to dispatch a sulet where dispat subet sees hey for the ad we do have a mapping so this meth is getting called so not just method is getting called you also getting the object of request that's a job of spring dependency injection now once you got the object using that

02:22:15
object you can fetch the values of I and J where we got three and five then we are adding them we are putting that data in a session okay so once you have dat a session we can fetch it any time in fact provided that session is running okay this is working but I still have some issues don't you think we are not using saet we are working with spring and this code looks bit ugly trust me it looks ugly can we clean it can we make it more effective and the answer is yes and that we'll see in the next video

02:22:49
[Music] okay so let's make this code look good the first issue is we don't want this HTTP request

object here don't you think spring should automatically give you this values in the parameter itself okay what I'm talking about is if I remove this part I want this two two values of int I and J here in I comma in J so we don't need to write these two lines okay so we are reled in the code right okay I'm just trying it out I'm not sure will it work okay the second thing is we

02:23:22
don't need object of HTP session here so yes we can actually pass it here so we can say HTTP session session so we don't have to say request.get session so our job is done right because if spring can give you HTTP request object it will also give you session object yes it it it does that okay but will this work of course it will not work because you are not sending I and J you are sending num one and num to but how will you assign it here so we have to use an annotation here which is at request param and here

02:23:55
in this request you have to mention whenever you send a value for n one assign the value for Num one to I is that simple okay let me import the package for request butam and I guess done and the same thing has to be done for J so let me just copy this code and paste it here okay and this we'll name it as num two so we got num one and num two so value of num one is going to I the value of num two is going to J so that's how you fetch the value okay this is actually coming from the address bar

02:14:33
you click on this button submit it will call add but don't you think we don't have anything here which will accept the request for add now think about this what can be done of course you need a page where you will show the output but before showing the output I want to process that information information so let's say if a user says 2 + 3 I want to give it five as answer where we do that so of course you need to create a controller so if you're coming from Ser background so what we do is we create

02:14:58
controllers but in Spring MVC you don't have to create a seate class every time what you can do is you can just add another method yes you can create a new class named it as calculator controller in which you can have different methods but as of now let's keep it here so in the same controller which is home controller we can have one more method so we'll do that I will say public this will return a string as of now let's keep it string and here I will say the method name as ADD of course you can

02:15:25
have any method name you you don't have to stick with ADD method name here and simply return the page you want to call of course right if you add these two numbers I want so on some page in fact instead of showing it on a page you can also return this as a data but we'll see that later how do we return a data as it is I want to call a page and we don't have a page as of now let's create one uh so we will just simply say copy and paste and I will name this page as result. JSP and in this JSP I don't want

02:15:53
the form and stuff I just want to print uh the output okay now how do we print the output so as of now simply thinging result is I'm not printing the actual output because we don't know the actual output let's go back here and in home controller you will simply say result. JSP that's it nothing much we have created two pages one is index.jsp and result. JSP okay so this looks fun okay I'm not sure if this is fun let me just try oh we're still missing something how do you know that you want

02:16:21
to call this add method for the add request of course it will not match the name you have to use a request mapping as The annotation So when you say request mapping in this you have to say add so whenever you call add request it will call add method is that simple let's run this so I will just restart the application once so let's go back to our code and let's refresh once again so when you ref now you can see not just we got welcome with teliscope but we also got two text field here let's say two

02:16:47
and three three as of now we have not done any coding for that let's click on submit okay you can see it is actually calling result page okay not in the address bar because if you see the address bar we have still have uh num 1 equal to two and num 2 equal to 3 okay uh we are not specifying which way is getting called here but result. JSP is getting called because we got the output as result is okay that looks cool but uh we want to perform the operation right so here I want to add those two numbers

02:17:15
now how will you do it now let's go back to our salet and GSP stuff uh if you have learned it before so in salet what we do is if you want to take the input from the user uh we use this very special interface which is known as sttp solet request so using that you can fetch values so what I will do I will just go to this ad and I will say hey I want to use those methods if remember it's not just one parameter we us to pass two parameters so in this service method we us to pass the request object

02:17:45
and the response object we don't have anything here so we have to create them right so let's do that so I will say HTTP Sate request and you can say req and we got that object so we'll say we we got this object now you'll be thinking how spring will know that you want the instance of this interface here that's the magic of spring dependencies injection right if you remember we have talked about spring ioc or spring dependency injection where it will inject the object which you need now

02:18:12
thankfully this object is available with Tom cat so spring will say okay we have that object in Tom let's use it that's great now once you got that object we can fetch the values now so I will say int I where you will have the first variable value I know it's not a logical name to use but that's fine so we'll simply say request. get parameter you have to also convert this into integer right so you have to say integer. parcent Okay that's how you do it in fact for second variable as well we do the same thing

02:18:40
let's copy this and paste and this time we'll name it as J of course and here I will say num two now once you got those two values you can simply add them right how do we add it uh it's very simple you can say int I will have the variable name as maybe num three I don't know if it makes sense I will say I + J so the value of I and J will be getting added to give you num three now I want to print this num three because this is my output in the result page oh how would I do that so we have to send this num

02:19:11
three to the result page and to do that it's very simple actually you can simply say Hey how do you

do that okay so in solet we used to use a concept of HTP session uh so when we use HTTP session we can simply add this num three in that object and in the JSP you can fetch it that's one way you can also use send disp where you can pass this value here you can say result. JSP question mark num3 equal to and whatever value of num three is you can say num three here that's URL rewriting right that's one

02:19:45
thing you can do we have multiple options right as of now what I will do I will simply say uh I will add it in a session so I will say HTTP session session is equal to Let's create the object of it oh but how do we do that so we okay first of all we have to say we have to say control space let's import the package and here I will say request dot get session so we have this method which is get session which will give you object of session okay once you have that you can simply say session dot you

02:20:15
can add not you can okay what's the method name I forgot the name you have to say set attribute and the attribute name we can have num three name can be anything right you can change it the only thing is whatever name you use here you have to use same name in the GSP page oh okay so will say num three in this case Okay once you got this attribute num three we are call result. JSP in result. JSP I just have to say uh we can use Java code here we can use session object and we can fetch the value or we can use jstl so as for the

02:20:46
jstl syntax we simply have have to say dollar and the cly packets and here you will say num three that's it so whatever name you have in the salet or in the Java code need to have it here okay it makes sense uh but but sometime sometime your gstl may not work so maybe your code is ignoring it so we will simply say is e ignored false or sometime it ignores e language which is expression language okay uh let's keep it just to be on safe side I'm saying false everything looks cool I think it will

02:21:19
work let's restart the application so let's go back to the page uh in fact I want to call the first page let's refresh in fact you can see the the address is Local Host colon 880 and then here I will say num one as three and num two as four or five uh so when you click on submit you want output as eight and we got it can you see that we got the output as eight so what is happening behind the scene uh so when you send a request the request is getting called for the homepage right so when you say

02:21:45
homepage this method is getting executed which is calling index.jsp and that's why we got that page which we have where we have welcome to telescope and then we have a form when you click on submit it is calling add request is again going to dispatch a sulet where dispat subet sees hey for the ad we do have a mapping so this meth is getting called so not just method is getting called you also getting the object of request that's a job of spring dependency injection now once you got the object using that

02:22:15
object you can fetch the values of I and J where we got three and five then we are adding them we are putting that data in a session okay so once you have dat a session we can fetch it any time in fact provided that session is running okay this is working but I still have some issues don't you think we are not using saet we are working with spring and this code looks bit ugly trust me it

looks ugly can we clean it can we make it more effective and the answer is yes and that we'll see in the next video

02:22:49
[Music] okay so let's make this code look good the first issue is we don't want this HTTP request object here don't you think spring should automatically give you this values in the parameter itself okay what I'm talking about is if I remove this part I want this two two values of int I and J here in I comma in J so we don't need to write these two lines okay so we are reled in the code right okay I'm just trying it out I'm not sure will it work okay the second thing is we

02:23:22
don't need object of HTP session here so yes we can actually pass it here so we can say HTTP session session so we don't have to say request.get session so our job is done right because if spring can give you HTTP request object it will also give you session object yes it it it does that okay but will this work of course it will not work because you are not sending I and J you are sending num one and num to but how will you assign it here so we have to use an annotation here which is at request param and here

02:23:55
in this request you have to mention whenever you send a value for n one assign the value for Num one to I is that simple okay let me import the package for request butam and I guess done and the same thing has to be done for J so let me just copy this code and paste it here okay and this we'll name it as num two so we got num one and num two so value of num one is going to I the value of num two is going to J so that's how you fetch the value okay this is actually coming from the address bar

02:24:21
we'll see that and then let's try this let's restart the application refresh the URL and you can see we got this page here I will say three and four now you can see this data is going to the server so let's me click on submit in the address bar you can see the values of three and four are going in the URL so it is switching from there and that's how that request P works that's great this is working yes you can also use uh post request year and it will still work so that's the benefit of using request

02:24:50
param so we are reducing the number of lines we are going for we don't want request object we don't need to assign we don't even have to pass it that's awesome right it it is working but still but still we don't want this HTTP session object here can we remove that we'll see that in the next [Music] video let's try to solve that let's remove that HTTP session here the way you can do that is with the help of a concept known as model and view okay how do we use that now if you talk about

02:25:24
model model will have data right we have discussed that in MVC so in MVC model is a data so your data here is this d three so we have to just we just have to put this num three in the model object and we also have to specify the view right and the view name is. JSP okay how do we do that so we have a inbuilt class model and view right we'll say MV we can have any name doesn't matter and we'll say new model and view now once you got this object here instead of using session we'll add this in a model view so I will

02:25:59
say MV dot there's the option of add object you just have to say add object Which object you want to add so I want to add num3 as num3 itself the way you do that with session right so we got

motor object the amazing thing is you don't even have to return this string format you can return MV here because MV will not just have the data but also the view name oh we have not assigned that let's do that so I will say MV do set view name and the view name is result. JSP remember it is view name because

02:26:31
most of the time people make mistakes here they say they they write set view it's actually set view name okay uh let me just remove the extra lines from here yeah so you can see we have added the value we have assigned that to a model object it's just that the return type here would be model and view that's it but then we don't need this thing here so we don't need model session object there our job is done okay let's try let's relaunch the application so let's go back to the page

02:27:02
and here let's refresh it it should work there's nothing which we have done extra click on submit and you can see we got eight so the power of model view is it will take two things the data which is your num three so we don't need HTTP session object and it will also have the view name so you don't have to return string format so this works and we are happy whatever we have done till now okay but we have some issues what if what if you want to have this page in different folder what if you don't want

02:27:33
to mention JSP in the code because you might want to change your view technology in future now you're using JSP in future you want to use something else can we make it Dynamic let's see that in the next video so till this point everything is working right we are able to add two numbers as well so if I say four and five I click on submit you got nine the only thing here is if you look at this code we are mentioning the extension as well right that's something we want to avoid because maybe in future I might be going....give important and teach me like this only in very very simple

Samiksha, I understand exactly what you want ❤️—**teach me extremely simply, step-by-step, just like the video**, with **small examples**, **breakdown**, **why**, **how**, and **what is happening internally**.

So let's take **your last concept**:

⭐ "How to give a default value using Spring Bean XML"

Example: **Age should not be 0, it should be 10 by default.**

🚀 SIMPLE EXPLANATION (LIKE YOU ARE A BEGINNER)

🧒 Imagine:

You have a class:

```java
public class Alien {
    private int age;

    public void setAge(int age) {
        this.age = age;
        System.out.println("Age assigned");
    }
    public void show() {
        System.out.println("Age is: " + age);
    }
}
```

If you DO NOTHING → age becomes **0** (because int default is 0).

🙅 But we DON'T want 0.

We want **default value = 10**.

There are **3 ways** to give this default value.

✅ WAY 1 — Directly initialize in class

```java
private int age = 10;
```

✔️ Simple
❌ Not recommended because value is "hard-coded".

✅ WAY 2 — Assign after creating object (in code)

```java
Alien a = new Alien();
a.setAge(10);
```

✔️ Works
❌ But bad because *Spring* should control object, not you.

💎 WAY 3 — BEST WAY (Spring XML Property Tag)

This is what you were learning.

In `applicationContext.xml`:

```xml
<bean id="alien" class="com.telusko.Alien">
    <property name="age" value="10" />
</bean>
```

What happens?

1️⃣ Spring sees **`<bean>`**
2️⃣ Spring creates object of Alien → calls **constructor**

3️⃣ Spring sees `<property name="age">`
4️⃣ Spring calls **setAge(10)**
5️⃣ Age becomes **10**, not 0 🎉

🧠 INTERNAL STEPS (VERY SIMPLE)

When Spring starts:

```vbnet
Step 1: Create object
Step 2: Call setAge(10)
Step 3: Done!
```

Output:

```csharp
Alien object created
Age assigned
Age is: 10
```

❗ IMPORTANT RULE

Property name must match setter name:

```csharp
property name="age"
→ Spring searches for setAge()
```

If setter is wrong:

```scss
setAge1()
```

Spring gives error:

❌ **Invalid property 'age'**

Because Spring is searching for:

```scss
setAge()
```

and it does NOT exist.

🎯 WHY WE USE PROPERTY TAG?

Because:

✔️ Spring should control object
✔️ Good design
✔️ Centralized configuration
✔️ Easy to change values later
✔️ Avoid hard-coding

🧩 NEXT CONCEPT YOU ASKED:

“What if property is NOT primitive (like int), but a CLASS?

Example: Alien HAS-A Laptop”

I can teach this also **step by step**, very simple, like above.

Just tell me:
👉 **Should I explain “reference type property injection using XML” next?**

or

👉 **Should I explain “Spring MVC add numbers example” next?**

Your choice ✨

02:28:07
for some other view technology maybe velocity or free marker in that case I don't want to mention jsb here that's one thing the second thing is if you look at the pages here all these pages are there in the web app folder now web app folder is actually a public folder which means anyone can access this page throughout the world you can they can simply go to web page and can then they can mention the website name slash the page name it will not even call the controller it will directly call the

02:28:32
page and it will work we don't want that right we want this pages to be secure that's one thing or maybe you want to keep it somewhere so that no one will know the path so if you want to make it secured one of the folder we can use here is web INF so you can keep these files in web INF they are private the only way to call them is through the controller and that's how you can make it private but what if you want to make it public but in a different folder that's what we do right we put all our

02:28:56
pages in different folders so let's say I will say new and I will create a folder here and the folder name is view or views and I want to put this file inside this view so if I drag it here that's done so we got that into views now and second thing I don't want to maintain the mention the extension here so let me remove that and remove this this one as well so you can see the the extension is removed and the path is also changed and after doing that if you restart the application and if you go back to

02:29:25

browser let's let's see if that works now if I click on submit you can see we got an error it says 44 so maybe it is not able to find something but what is that something if I go back here so that something is the page right so of course it is not able to find this result. JSP now in this case you have to mention to your framework hey uh we are using spring framework and then spring framework is using disp servet so you have to inform this spet Hey whenever you are looking for the view you have to

02:29:52
find that in this folder which is views and then you have to also add the exchange of. JSP of course it should be configurable to configure this we have a very special file here which is application. properties this is where you will do the configuration but how because when you say property files we have to mention the key and a value so of course values we know it will be the part of the JSP which is views and the extension which is JSP but we don't the the property of it right and so normally

02:30:19
when you work on the project you do remember those things but if what if you are doing for this for the first time in that case you can simply go to this website which is the first one if you search for spring boot application properties go to this website and this is where you will find all the properties if you can just go down you can see we have this properties for logging we have aop we have identity then admin banners you can also change the banner right using this what we want is spring MVC so if you can just scroll

02:30:45
down so we got spring M you can see that and then here if you want to do that so if you want to assign the path and the extension we have to use this two things the prefix and suffix now prefix is what before result which is the path and suffix is what after result okay so which result I'm talking about so if you go back to the code uh you can see here we are mentioning result right so before that we have to mention views after that we have to mention extension and to achieve that we have to put those two

02:31:15
properties here so we can go back there we can copy these two properties and we can paste it here of course we have to change this so what should be the prefix so prefix would be views of course right that makes sense uh so when you search for the file now it will be views slash and the file name we have to also mention the extension so you have to say dot JSP that's it once you have done that I hope this time this will work let's restart the application and let's go back to the browser and here if I

02:31:42
refresh this and if I click on submit oh it's working can you see that so that's how you can achieve you achieve this thing which is removing the extension which is JSP from the code from the Java code and putting that into in the property files the second thing is we have to also change the views in fact if you want to make it private you can also say views belongs to web in folder and that will make it private so that's how you achieve prefix and suffix see you in the next [Music]

02:32:10
video so till this point everything is working right so we have added these two numbers in fact the numbers which I'm talking about is this so if I say if I pass let's say three and four and if I click on submit you can see we got the output as result is seven so that's what we wanted to achieve that what we are doing is we have created a controller so whenever you request for homepage it will

return the index.jsp and then when you request for so in fact in the index.jsp if you go there what we are doing is when you

02:32:41
click on submit it is sending a request for add oh so where we are handling it so you can see in the controller itself we have ADD method which accept the request for ADD and everything is working you can see it says request Pam we got num one num two and things are working okay but there's one thing which we have done in the last video which is the introduction of model and view so the reason we went for this is uh when you want to call a view of course right you want to send some data to a client

02:33:12
and client will expect a HTML page so what we are doing is we have Creed the JSP which will get converted into HTML after rendering but then when we say you want to call a JSP JSP also need a data and data in this case is this n three so basically we have to set up two things the first one is the view name and the second one is the data which you want to pass so data in this case is the num three okay so for that we are using this model and view because it can hold two things model the data and the view name

02:33:46
which is uh result in this case result. JSP so what I want to achieve now is do we have some alternate option example let's say this is best model view is best we can set the name in fact there's one more thing about model view you can uh actually set the view name here itself so you don't have to uh use another method so you can simply pass the view name in The Constructor and that works okay so the point is can we use something else apart from model and view example let's say I don't want to

02:24:21
we'll see that and then let's try this let's restart the application refresh the URL and you can see we got this page here I will say three and four now you can see this data is going to the server so let's me click on submit in the address bar you can see the values of three and four are going in the URL so it is switching from there and that's how that request P works that's great this is working yes you can also use uh post request year and it will still work so that's the benefit of using request

02:24:50
param so we are reducing the number of lines we are going for we don't want request object we don't need to assign we don't even have to pass it that's awesome right it it is working but still but still we don't want this HTTP session object here can we remove that we'll see that in the next [Music] video let's try to solve that let's remove that HTTP session here the way you can do that is with the help of a concept known as model and view okay how do we use that now if you talk about

02:25:24
model model will have data right we have discussed that in MVC so in MVC model is a data so your data here is this d three so we have to just we just have to put this num three in the model object and we also have to specify the view right and the view name is. JSP okay how do we do that so we have a inbuilt class model and view right we'll say MV we can have any name doesn't matter and we'll say new model and view now once you got this object here instead of using session we'll add this in a model view so I will

02:25:59
say MV dot there's the option of add object you just have to say add object Which object you want

to add so I want to add num3 as num3 itself the way you do that with session right so we got motor object the amazing thing is you don't even have to return this string format you can return MV here because MV will not just have the data but also the view name oh we have not assigned that let's do that so I will say MV do set view name and the view name is result. JSP remember it is view name because

02:26:31
most of the time people make mistakes here they say they they write set view it's actually set view name okay uh let me just remove the extra lines from here yeah so you can see we have added the value we have assigned that to a model object it's just that the return type here would be model and view that's it but then we don't need this thing here so we don't need model session object there our job is done okay let's try let's relaunch the application so let's go back to the page

02:27:02
and here let's refresh it it should work there's nothing which we have done extra click on submit and you can see we got eight so the power of model view is it will take two things the data which is your num three so we don't need HTTP session object and it will also have the view name so you don't have to return string format so this works and we are happy whatever we have done till now okay but we have some issues what if what if you want to have this page in different folder what if you don't want

02:27:33
to mention JSP in the code because you might want to change your view technology in future now you're using JSP in future you want to use something else can we make it Dynamic let's see that in the next video so till this point everything is working right we are able to add two numbers as well so if I say four and five I click on submit you got nine the only thing here is if you look at this code we are mentioning the extension as well right that's something we want to avoid because maybe in future I might be going

02:28:07
for some other view technology maybe velocity or free marker in that case I don't want to mention jsb here that's one thing the second thing is if you look at the pages here all these pages are there in the web app folder now web app folder is actually a public folder which means anyone can access this page throughout the world you can they can simply go to web page and can then they can mention the website name slash the page name it will not even call the controller it will directly call the

02:28:32
page and it will work we don't want that right we want this pages to be secure that's one thing or maybe you want to keep it somewhere so that no one will know the path so if you want to make it secured one of the folder we can use here is web INF so you can keep these files in web INF they are private the only way to call them is through the controller and that's how you can make it private but what if you want to make it public but in a different folder that's what we do right we put all our

02:28:56
pages in different folders so let's say I will say new and I will create a folder here and the folder name is view or views and I want to put this file inside this view so if I drag it here that's done so we got that into views now and second thing I don't want to maintain the mention the extension here so let me remove that and remove this this one as well so you can see the the extension is removed and the path is also changed and after doing that if you restart the application and if you go back

to

02:29:25
browser let's let's see if that works now if I click on submit you can see we got an error it says 44 so maybe it is not able to find something but what is that something if I go back here so that something is the page right so of course it is not able to find this result. JSP now in this case you have to mention to your framework hey uh we are using spring framework and then spring framework is using disp servet so you have to inform this spet Hey whenever you are looking for the view you have to

02:29:52
find that in this folder which is views and then you have to also add the exchange of. JSP of course it should be configurable to configure this we have a very special file here which is application. properties this is where you will do the configuration but how because when you say property files we have to mention the key and a value so of course values we know it will be the part of the JSP which is views and the extension which is JSP but we don't the the property of it right and so normally

02:30:19
when you work on the project you do remember those things but if what if you are doing for this for the first time in that case you can simply go to this website which is the first one if you search for spring boot application properties go to this website and this is where you will find all the properties if you can just go down you can see we have this properties for logging we have aop we have identity then admin banners you can also change the banner right using this what we want is spring MVC so if you can just scroll

02:30:45
down so we got spring M you can see that and then here if you want to do that so if you want to assign the path and the extension we have to use this two things the prefix and suffix now prefix is what before result which is the path and suffix is what after result okay so which result I'm talking about so if you go back to the code uh you can see here we are mentioning result right so before that we have to mention views after that we have to mention extension and to achieve that we have to put those two

02:31:15
properties here so we can go back there we can copy these two properties and we can paste it here of course we have to change this so what should be the prefix so prefix would be views of course right that makes sense uh so when you search for the file now it will be views slash and the file name we have to also mention the extension so you have to say dot JSP that's it once you have done that I hope this time this will work let's restart the application and let's go back to the browser and here if I

02:31:42
refresh this and if I click on submit oh it's working can you see that so that's how you can achieve you achieve this thing which is removing the extension which is JSP from the code from the Java code and putting that into in the property files the second thing is we have to also change the views in fact if you want to make it private you can also say views belongs to web in folder and that will make it private so that's how you achieve prefix and suffix see you in the next [Music]

02:32:10
video so till this point everything is working right so we have added these two numbers in fact the

numbers which I'm talking about is this so if I say if I pass let's say three and four and if I click on submit you can see we got the output as result is seven so that's what we wanted to achieve that what we are doing is we have created a controller so whenever you request for homepage it will return the index.jsp and then when you request for so in fact in the index.jsp if you go there what we are doing is when you

02:32:41
click on submit it is sending a request for add oh so where we are handling it so you can see in the controller itself we have ADD method which accept the request for ADD and everything is working you can see it says request Pam we got num one num two and things are working okay but there's one thing which we have done in the last video which is the introduction of model and view so the reason we went for this is uh when you want to call a view of course right you want to send some data to a client

02:33:12
and client will expect a HTML page so what we are doing is we have Creed the JSP which will get converted into HTML after rendering but then when we say you want to call a JSP JSP also need a data and data in this case is this n three so basically we have to set up two things the first one is the view name and the second one is the data which you want to pass so data in this case is the num three okay so for that we are using this model and view because it can hold two things model the data and the view name

02:33:46
which is uh result in this case result. JSP so what I want to achieve now is do we have some alternate option example let's say this is best model view is best we can set the name in fact there's one more thing about model view you can uh actually set the view name here itself so you don't have to uh use another method so you can simply pass the view name in The Constructor and that works okay so the point is can we use something else apart from model and view example let's say I don't want to

02:34:17
return model view object here I I want to stick to the string part here okay uh that means in the return we have to still mention the view name so I will say result what I have to change is I don't want to use model view so then the question arise how will you add the object that's what we want to do right we want to add this data in the model in that case if you simply want a model you can create an object of model and it will still work but where so the idea is I want want to put this data in the

02:34:48
model object so model is something which will be available for the view as well so in the parameter in the bracket itself you can say you can specify the reference of model M okay now what this model is so model is basically an interface which will behave same as model view except that we don't have a view here so you can say we are saying model and here instead of saying MV dot we will be saying m dot but you can see we got an error the thing is in M we don't have add object method we have to

02:35:18
use another method which is quite similar you can see the method name is ADD attribute so instead of saying add object we have to say add attribute and then we can pass the two stuff the same thing as model view uh it's just that we are not specifying The View separately now which one to use it depend upon you you can still go with model or you can go with model view it's just that

when you send a request and if you want to have that data in a model doesn't matter if it would be used for view or not or

02:35:47
you want to call a specific View later you can still use model but if you want to specify model view both I would say preferably you can go with model and view okay you still have a choice that's great that's how we can use model here now other than this model we also have an option of model map so you can say model map here even that works you can say control space and enter you can see the this the code will still work so let me just run this code and let we also see the difference between these two so

02:36:13
let me first run the code with model map uh let me relaunch the application and it is still launching so let's refresh and you can see we got the same page there's no error that's great this time we'll add some different numbers we'll say 6 and 8 when you click on submit you got the answer so you can see we have not done any changes and at the output and the only change we have done is in the controller so instead of using model and view we are using model map then what's the difference between model and

02:36:42
model map so if you click on this you can see this is documentation for model map model map as name suggest supports a feature of map so by default when you add the attribute will get added in the map format so you are adding data in a map which is getting added in the model object so it depend upon you which one to use if you want to use Simple object I would say go for model if you want to add map if you want to have the data in a map format of course the best option would be model and map I me model map

02:37:10
okay let's go back here so this still works with model and model map both at least this code so try it out uh so it will it will of course work so that's it from this video where we have talked about model and model [Music] map now once we have talked about model and model map it's time to look at another annotation which is model attribute now before doing that I want to make some changes the first change is instead of adding two numbers let's do something else now in Java everything is

02:37:42
object right so if you want to represent data we represent data in the object format example if you want to represent a laptop if you want to represent a mobile phone or maybe Account Details or user details everything will become a class and then we instantiate them right that's what we do in Java so here as well I want to create an object of alien okay now what is alien basically so alien is a programmer so I refer programmers as alien so what I will do here is I will go back here and I will

02:38:09
say new I want to create a model class so let's say class and we'll name this class as a alien and the only thing is I just want to make this into a different package okay so here I will say the package is model it's better to have all your model classes in model package now this alien will have certain variables right so let's say I want to go only for two variables here so I'll say private in a ID and private string a name of course in future we'll add some more attributes once we start talking about

02:38:41
Advance concept as of now these two are enough and now I want to have gets and sets for this I

will say Source generate gets and sets select all and generate so once we got the class once we got the variables and geta sets done let's go back to index.jsp now what I want is as a user or as a user you will add your own records uh so you'll be having the option of adding the aliens so of course the I want to have the action as ADD itself but then the data which you will ask is different so we'll say enter

02:39:15
enter your ID so we're asking for the ID and of course this name will change I will say this is Aid and then here I will say enter your name so we're asking for two things a ID and name let's have it a name so you can see we got this two details so now when you click on submit it will send the add request in fact then just to make it more interactive I will say add alien it makes much more sense right so now when you click on submit it will call add alien but in the home controller you can see we don't

02:39:46
don't have anything as ADD alien so we'll modify this and we'll make it alien in fact you know what I will do instead of changing the existing stuff let me add another method here and say okay let me just say this is so I will say public string and the method name would be add alien of course you can go with any method name and now to call this I have to map it a request mapping right so I will say request mapping and here I will say add alien now we have to create this method now of course we have

02:40:15
to accept some parameters right so the parameter which we are passing from the index.jsp is these two details we have a ID and a name so whatever we have learn till now is uh if you want to accept the values we have to use this two annotation here in fact two variables with two annotations we have to say request param so we can accept that with the help of request param the only thing is here it will say a ID and the variable name is also a ID and here I will say instead of num two we'll say a

02:40:43
name and here as well I will say a name okay there's one change a name is not of type int it's of type string so they have to take care of that as well so that's done now this method will return something right I want to call a page I want to call the same result page I will say result uh because I'm lazy to create one more page so let's stick with that of course you can create your own Pages later but as of now I want to call result but this result page need to have something right I want to print

02:41:10
something here so what I will do is I will not print num three I will print the object of a I want to print alien here but will this work of course if you go back to home controller we are calling result that will work but then it will not print alien because we don't have that object in the model as you can see we don't have any model in this method so the first step is you need a model object so I will simply give a comma here and I will say model M so that at least I will be having the model

02:41:37
object with me and in that model object I want to pass the object of alien the amazing thing is we don't have the object yet so let's create one and the way you create object you you can say alien a equal to new alien and then here you will say let import the package and here you will say a DOT we have to assign the values as well right whatever is coming from the client so as a client you will be sending data right so we will say submit by passing let's say a ID is 101 and name as naen so

this two

02:42:07
details will be received here I mean here in this two variables we have to assign that to the object so we say a do set a name I will say set a name as a name in fact once you got the object we have to add this object in the model object so I will say m do add attribute and here you will simply add the attribute as a but if you add the attribute as a it will not work because in the result. JSP we are asking for the alien so the name we have to refer here is alien so I would say alien so whatever data is there in uh in a will

02:42:41
be represented as alien in the result. jsb that's what we want now will this work see till this point we have not done anything extra it's just that we got a class in fact whatever we have done till now we know about this stuff let's relaunch the application just to verify if everything is working so application has been launched let's go back to the page let's refresh so yes of course we need to see some changes instead of having uh num one num two we'll be getting ID and name so I will

02:43:07
say the ID is 101 and name is naen I guess I missed something in the code let's see that what I missed when you click on submit you can see we are getting the the default uh package name class name and the hash code we don't want this we want to print a data right to achieve that we do one thing right we always create we override the two string method if remember in normal code Java we do that so I will simply run generate a two string method here which will print data this is what I missed let me

02:43:36
just save it and let's relaunch the application again so when you click on submit here you can see we got the output so this is what we wanted right so whatever user has entered will be coming here here now you will say hey everything is working then what's new here the thing is if you look at this code which is request param and request bam something is not matching right we are accepting individual values and then we are assigning them to the object here it doesn't look good right what I want

02:34:17
return model view object here I I want to stick to the string part here okay uh that means in the return we have to still mention the view name so I will say result what I have to change is I don't want to use model view so then the question arise how will you add the object that's what we want to do right we want to add this data in the model in that case if you simply want a model you can create an object of model and it will still work but where so the idea is I want want to put this data in the

02:34:48
model object so model is something which will be available for the view as well so in the parameter in the bracket itself you can say you can specify the reference of model M okay now what this model is so model is basically an interface which will behave same as model view except that we don't have a view here so you can say we are saying model and here instead of saying MV dot we will be saying m dot but you can see we got an error the thing is in M we don't have add object method we have to

02:35:18
use another method which is quite similar you can see the method name is ADD attribute so

instead of saying add object we have to say add attribute and then we can pass the two stuff the same thing as model view uh it's just that we are not specifying The View separately now which one to use it depend upon you you can still go with model or you can go with model view it's just that when you send a request and if you want to have that data in a model doesn't matter if it would be used for view or not or

02:35:47
you want to call a specific View later you can still use model but if you want to specify model view both I would say preferably you can go with model and view okay you still have a choice that's great that's how we can use model here now other than this model we also have an option of model map so you can say model map here even that works you can say control space and enter you can see the this the code will still work so let me just run this code and let we also see the difference between these two so

02:36:13
let me first run the code with model map uh let me relaunch the application and it is still launching so let's refresh and you can see we got the same page there's no error that's great this time we'll add some different numbers we'll say 6 and 8 when you click on submit you got the answer so you can see we have not done any changes and at the output and the only change we have done is in the controller so instead of using model and view we are using model map then what's the difference between model and

02:36:42
model map so if you click on this you can see this is documentation for model map model map as name suggest supports a feature of map so by default when you add the attribute will get added in the map format so you are adding data in a map which is getting added in the model object so it depend upon you which one to use if you want to use Simple object I would say go for model if you want to add map if you want to have the data in a map format of course the best option would be model and map I me model map

02:37:10
okay let's go back here so this still works with model and model map both at least this code so try it out uh so it will it will of course work so that's it from this video where we have talked about model and model [Music] map now once we have talked about model and model map it's time to look at another annotation which is model attribute now before doing that I want to make some changes the first change is instead of adding two numbers let's do something else now in Java everything is

02:37:42
object right so if you want to represent data we represent data in the object format example if you want to represent a laptop if you want to represent a mobile phone or maybe Account Details or user details everything will become a class and then we instantiate them right that's what we do in Java so here as well I want to create an object of alien okay now what is alien basically so alien is a programmer so I refer programmers as alien so what I will do here is I will go back here and I will

02:38:09
say new I want to create a model class so let's say class and we'll name this class as a alien and the only thing is I just want to make this into a different package okay so here I will say the package is model it's better to have all your model classes in model package now this alien will have certain variables right so let's say I want to go only for two variables here so I'll say private in a ID and private string a name of course in future we'll add some more attributes once we start talking about

02:38:41
Advance concept as of now these two are enough and now I want to have gets and sets for this I will say Source generate gets and sets select all and generate so once we got the class once we got the variables and geta sets done let's go back to index.jsp now what I want is as a user or as a user you will add your own records uh so you'll be having the option of adding the aliens so of course the I want to have the action as ADD itself but then the data which you will ask is different so we'll say enter

02:39:15
enter your ID so we're asking for the ID and of course this name will change I will say this is Aid and then here I will say enter your name so we're asking for two things a ID and name let's have it a name so you can see we got this two details so now when you click on submit it will send the add request in fact then just to make it more interactive I will say add alien it makes much more sense right so now when you click on submit it will call add alien but in the home controller you can see we don't

02:39:46
don't have anything as ADD alien so we'll modify this and we'll make it alien in fact you know what I will do instead of changing the existing stuff let me add another method here and say okay let me just say this is so I will say public string and the method name would be add alien of course you can go with any method name and now to call this I have to map it a request mapping right so I will say request mapping and here I will say add alien now we have to create this method now of course we have

02:40:15
to accept some parameters right so the parameter which we are passing from the index.jsp is these two details we have a ID and a name so whatever we have learn till now is uh if you want to accept the values we have to use this two annotation here in fact two variables with two annotations we have to say request param so we can accept that with the help of request param the only thing is here it will say a ID and the variable name is also a ID and here I will say instead of num two we'll say a

02:40:43
name and here as well I will say a name okay there's one change a name is not of type int it's of type string so they have to take care of that as well so that's done now this method will return something right I want to call a page I want to call the same result page I will say result uh because I'm lazy to create one more page so let's stick with that of course you can create your own Pages later but as of now I want to call result but this result page need to have something right I want to print

02:41:10
something here so what I will do is I will not print num three I will print the object of a I want to print alien here but will this work of course if you go back to home controller we are calling result that will work but then it will not print alien because we don't have that object in the model as you can see we don't have any model in this method so the first step is you need a model object so I will simply give a comma here and I will say model M so that at least I will be having the model

02:41:37
object with me and in that model object I want to pass the object of alien the amazing thing is we don't have the object yet so let's create one and the way you create object you you can say alien a

equal to new alien and then here you will say let import the package and here you will say a DOT we have to assign the values as well right whatever is coming from the client so as a client you will be sending data right so we will say submit by passing let's say a ID is 101 and name as naen so this two

02:42:07
details will be received here I mean here in this two variables we have to assign that to the object so we say a do set a name I will say set a name as a name in fact once you got the object we have to add this object in the model object so I will say m do add attribute and here you will simply add the attribute as a but if you add the attribute as a it will not work because in the result. JSP we are asking for the alien so the name we have to refer here is alien so I would say alien so whatever data is there in uh in a will

02:42:41
be represented as alien in the result. jsb that's what we want now will this work see till this point we have not done anything extra it's just that we got a class in fact whatever we have done till now we know about this stuff let's relaunch the application just to verify if everything is working so application has been launched let's go back to the page let's refresh so yes of course we need to see some changes instead of having uh num one num two we'll be getting ID and name so I will

02:43:07
say the ID is 101 and name is naen I guess I missed something in the code let's see that what I missed when you click on submit you can see we are getting the the default uh package name class name and the hash code we don't want this we want to print a data right to achieve that we do one thing right we always create we override the two string method if remember in normal code Java we do that so I will simply run generate a two string method here which will print data this is what I missed let me

02:43:36
just save it and let's relaunch the application again so when you click on submit here you can see we got the output so this is what we wanted right so whatever user has entered will be coming here here now you will say hey everything is working then what's new here the thing is if you look at this code which is request param and request bam something is not matching right we are accepting individual values and then we are assigning them to the object here it doesn't look good right what I want

02:44:03
is instead of having request but I'm here what if I can simply mention the alien object here itself this is what we wanted that means I don't have to mention all this thing so the moment you pass the object on the moment you pass the values you know in a ID and a name now since the names are same right it's in a ID and a name don't you think it should be assigned automatically to the alen object is it possible yes and the way you can do that is with the help of model attribute so once you have mod

02:44:32
attribute with you you can directly assign the values here but how to do that that we'll see in the next video so let's make it work to make it work we have have to use another annotation which is model attribute so that means here instead of this all this request param this stuff we don't need this a ID and a name separately what I want is I want to say alien a right this is this is what we wanted right and then to achieve that we have to we don't need this stuff here we need to remove that

02:45:06
the only thing is we want this object to be received from the client to achieve that we have to use an annotation here which is ADD model attribute that's it simply save this so whatever is coming from the client will be assigned to this a and our job is done let's verify let's relaunch the application so this time I'm sending one2 let's click on submit and you can see we got the data this is what we wanted right and the amazing thing is on the client side I mean in the code we are not using two request

02:45:37
Pam annotation and then checking it one by one creating the object of alen nothing like that it's very simple right the amazing thing is model attribute is also responsible to add the data in the model that means we don't have to do this manually whatever object you got is by default there in the model and that's why it says model attribute right so we don't even need that model object it's quite simple okay but what if if I go back to result you can see here we are saying alien right and here we are

02:46:05
saying a what I will do now is let's go back here and here I will say A1 will will this work so I'm not using alien I'm not using a because a is the name of the attribute and alien is the class name itself or you can say the decapitalized class name but here we have a different name we have A1 as a name in this case let's try what will happen so let's click on submit the moment you do that you can see we have not received the data there the reason is we have a different name here which

02:46:33
is a and then here we are saying A1 okay how do we solve that so model attribute says okay if you want a different name that's fine in the bracket you can mention A1 so now whatever data is coming from this a will be assigned to A1 is that simple let's relaunch so when you click on submit here you can see we got the data so that's the beauty about model attribute so it's an annotation which also accept the parameters so if you pass the parameters it will take that name otherwise it will take the

02:47:01
name of the class which is alien okay but what if I have alien here okay let's stick to alien and here in the home controller what if I what if I don't write model attribute will this work the answer is yes if you don't have a different name if you still want to go with alien you don't have to mention model attribute but it's a it's a convention that you should write it otherwise it will still work but then you know there will be a confusion because here we are saying alien uh so

02:47:29
it is working what if you change it what if you say A1 it will not work so let's relaunch it without using mod attribute it will still work so when you click on submit you can say it is still working so without even without M attribute it works because we are using the same name in as alien here here okay now this is mod attribute which we have used in the as a parameter to the method can we use mod attribute at the method level we'll we'll see that in the next video so let's say here in the alien I

02:48:03
want one more data so let me create one more class and I want some data about it or maybe not even class let's say I want to print the message let's say welcome message I will say welcome back aliens so I want to print this message the one way you can do that is by printing as it is or maybe I

want to change this alien to something else every time I maybe maybe I want to I want to do that in the code so what we can do is we can say welcome back and here we'll say name so by default you can see we don't have any

02:48:31
name here right so name is just an attribute which we are using here that means before calling result I have to assign name where should I do that of course you will say you have to do that in this method because this is what is calling result but what if you have multiple methods which is calling result in this case you can see even this method is calling result so of course we should not be assigning the name here what you can do is you can create a separate method a separate method which was responsible to assign the value to

02:48:57
the name so let's say I will say public void I will say model data of course you can have a name doesn't matter and here you will simply assign the data to the model okay so how do we do that so we have to use a model M now in this model you can say m dot and you can say m add attribute of course with the name uh so we'll say the name here is name and then the actual name which I want to pass is aliens of course you can pass anything here but let me just pass aliens now what will happen is in the code in the

02:49:28
output I want to welcome back aliens right so the name will be replaced by aliens but will this work of course not if you want to make it work we have to use and annotation here which is model attribute the moment you say model attribute what will happen is your spring framework before calling any request mapping it will first call model attribute this model attribute will create a model object and then in that model object you are adding the attribute which is name with the name aliens now it doesn't matter whatever

02:49:57
controller you call whatever method you call with the request mapping everywhere this name will be available so in the result page you will get that name right so we can use motor attribute in the method parameter or on the method level so let's relaunch it so let's click on submit here and you can see we got result is okay we have not received the result I don't know why what went wrong there but at least it is working you can see we got welcome back aliens so let's figure it out what went wrong there why

02:50:24
we have not received the alien okay because we are saying alien here and the attribute here is A1 that's bad uh let's remove that I guess that was the issue let's save that okay but at least that this is getting right we are getting welcome back aliens at is that is working uh let's relaunch it okay so let's click on submit and so that that is working we got a result is this so we got this data and then we also got welcome back aliens so that's how we can use M attribute on method

02:50:51
[Music] level so whatever we have done till now is awesome right uh we have created this project and it works like charm so we can send data to the server and server will give you the response the amazing thing is instead of using normal Ser JSP we are using a spring MVC framework and that to we are not using normal spring MVC we are using spring boot here the reason we are using spring boot is when you work on Spring MVC project or any type of spring projects we need to do a lot of configuration and spring boot

02:51:24
says hey you don't have to worry about the configuration so if you look at the project we have not done much of the configuration we were focusing only on the logic yes we did some configuration in the property file so if I open that file for you so you can see this is where we are mentioned that we want to work with this folder views and then the extension for the uh page is JSP other than that everything is quite simple right we are doing everything in the Cod but then before spring board we should

02:51:52
still work on Spring MVC I mean a normal Maven project where you have to do a lot of configuration until this point we have not seen that so what if you are working on a project of spring MVC but without spring boot so in this video Let's create a project with the same features I want to have alien I want to have homepage so with the same feature but with the help of normal spring MVC configuration so yes we will be using mavin here we have to add dependencies we have to do some configuration so what

02:52:23
config I'm talking about now first of all when you say spring boot project so this project knows that this is spring boot MVC so whenever you send a request request directly goes to dispatch s but when you create a normal mavin project or normal web project we have to navigate we have to mention hey if you want send your request first to dispat a salet so we have to configure that second thing you have to configure is we have to mention that this is a spring MVC project where all the configuration

02:52:52
I mean all the controller have annotations okay that's we have to mention second we have to do this configuration as well but then what type of file we'll be using to do this configuration so in general when you talk about configuration one thing come to mind is XML so we'll be using XML to configure this okay it looks lot of work right and technically yes we have to do a lot of work but it's it's worth learning so what I will do is I will go back to my application I mean go back to

02:53:21
the project and instead of using the same project I'll be creating my own I mean I'll be getting a new project here so let me right click here and we send new and of course in the new project I will be just copy pasting the existing code to save our time so here I will create a maven project okay so not a spring project or something let's create a m project and in this MAV project so click on next I want to create a web application select that click on next so it is asking for the group ID here so I will

02:44:03
is instead of having request but I'm here what if I can simply mention the alien object here itself this is what we wanted that means I don't have to mention all this thing so the moment you pass the object on the moment you pass the values you know in a ID and a name now since the names are same right it's in a ID and a name don't you think it should be assigned automatically to the alen object is it possible yes and the way you can do that is with the help of model attribute so once you have mod

02:44:32
attribute with you you can directly assign the values here but how to do that that we'll see in the next video so let's make it work to make it work we have have to use another annotation which is model attribute so that means here instead of this all this request param this stuff we don't need

this a ID and a name separately what I want is I want to say alien a right this is this is what we wanted right and then to achieve that we have to we don't need this stuff here we need to remove that

02:45:06
the only thing is we want this object to be received from the client to achieve that we have to use an annotation here which is ADD model attribute that's it simply save this so whatever is coming from the client will be assigned to this a and our job is done let's verify let's relaunch the application so this time I'm sending one2 let's click on submit and you can see we got the data this is what we wanted right and the amazing thing is on the client side I mean in the code we are not using two request

02:45:37
Pam annotation and then checking it one by one creating the object of alen nothing like that it's very simple right the amazing thing is model attribute is also responsible to add the data in the model that means we don't have to do this manually whatever object you got is by default there in the model and that's why it says model attribute right so we don't even need that model object it's quite simple okay but what if if I go back to result you can see here we are saying alien right and here we are

02:46:05
saying a what I will do now is let's go back here and here I will say A1 will will this work so I'm not using alien I'm not using a because a is the name of the attribute and alien is the class name itself or you can say the decapitalized class name but here we have a different name we have A1 as a name in this case let's try what will happen so let's click on submit the moment you do that you can see we have not received the data there the reason is we have a different name here which

02:46:33
is a and then here we are saying A1 okay how do we solve that so model attribute says okay if you want a different name that's fine in the bracket you can mention A1 so now whatever data is coming from this a will be assigned to A1 is that simple let's relaunch so when you click on submit here you can see we got the data so that's the beauty about model attribute so it's an annotation which also accept the parameters so if you pass the parameters it will take that name otherwise it will take the

02:47:01
name of the class which is alien okay but what if I have alien here okay let's stick to alien and here in the home controller what if I what if I don't write model attribute will this work the answer is yes if you don't have a different name if you still want to go with alien you don't have to mention model attribute but it's a it's a convention that you should write it otherwise it will still work but then you know there will be a confusion because here we are saying alien uh so

02:47:29
it is working what if you change it what if you say A1 it will not work so let's relaunch it without using mod attribute it will still work so when you click on submit you can say it is still working so without even without M attribute it works because we are using the same name in as alien here here okay now this is mod attribute which we have used in the as a parameter to the method can we use mod attribute at the method level we'll we'll see that in the next video so let's say here in the alien I

02:48:03

want one more data so let me create one more class and I want some data about it or maybe not even class let's say I want to print the message let's say welcome message I will say welcome back aliens so I want to print this message the one way you can do that is by printing as it is or maybe I want to change this alien to something else every time I maybe maybe I want to I want to do that in the code so what we can do is we can say welcome back and here we'll say name so by default you can see we don't have any

02:48:31
name here right so name is just an attribute which we are using here that means before calling result I have to assign name where should I do that of course you will say you have to do that in this method because this is what is calling result but what if you have multiple methods which is calling result in this case you can see even this method is calling result so of course we should not be assigning the name here what you can do is you can create a separate method a separate method which was responsible to assign the value to

02:48:57
the name so let's say I will say public void I will say model data of course you can have a name doesn't matter and here you will simply assign the data to the model okay so how do we do that so we have to use a model M now in this model you can say m dot and you can say m add attribute of course with the name uh so we'll say the name here is name and then the actual name which I want to pass is aliens of course you can pass anything here but let me just pass aliens now what will happen is in the code in the

02:49:28
output I want to welcome back aliens right so the name will be replaced by aliens but will this work of course not if you want to make it work we have to use and annotation here which is model attribute the moment you say model attribute what will happen is your spring framework before calling any request mapping it will first call model attribute this model attribute will create a model object and then in that model object you are adding the attribute which is name with the name aliens now it doesn't matter whatever

02:49:57
controller you call whatever method you call with the request mapping everywhere this name will be available so in the result page you will get that name right so we can use motor attribute in the method parameter or on the method level so let's relaunch it so let's click on submit here and you can see we got result is okay we have not received the result I don't know why what went wrong there but at least it is working you can see we got welcome back aliens so let's figure it out what went wrong there why

02:50:24
we have not received the alien okay because we are saying alien here and the attribute here is A1 that's bad uh let's remove that I guess that was the issue let's save that okay but at least that this is getting right we are getting welcome back aliens at is that is working uh let's relaunch it okay so let's click on submit and so that that is working we got a result is this so we got this data and then we also got welcome back aliens so that's how we can use M attribute on method

02:50:51
[Music] level so whatever we have done till now is awesome right uh we have created this project and it works like charm so we can send data to the server and server will give you the response the amazing thing is instead of using normal Ser JSP we are using a spring MVC framework and that to we are not using normal spring MVC we are using spring boot here the reason we are using spring

boot is when you work on Spring MVC project or any type of spring projects we need to do a lot of configuration and spring boot

02:51:24
says hey you don't have to worry about the configuration so if you look at the project we have not done much of the configuration we were focusing only on the logic yes we did some configuration in the property file so if I open that file for you so you can see this is where we are mentioned that we want to work with this folder views and then the extension for the uh page is JSP other than that everything is quite simple right we are doing everything in the Cod but then before spring board we should

02:51:52
still work on Spring MVC I mean a normal Maven project where you have to do a lot of configuration until this point we have not seen that so what if you are working on a project of spring MVC but without spring boot so in this video Let's create a project with the same features I want to have alien I want to have homepage so with the same feature but with the help of normal spring MVC configuration so yes we will be using mavin here we have to add dependencies we have to do some configuration so what

02:52:23
config I'm talking about now first of all when you say spring boot project so this project knows that this is spring boot MVC so whenever you send a request request directly goes to dispatch s but when you create a normal mavin project or normal web project we have to navigate we have to mention hey if you want send your request first to dispat a salet so we have to configure that second thing you have to configure is we have to mention that this is a spring MVC project where all the configuration

02:52:52
I mean all the controller have annotations okay that's we have to mention second we have to do this configuration as well but then what type of file we'll be using to do this configuration so in general when you talk about configuration one thing come to mind is XML so we'll be using XML to configure this okay it looks lot of work right and technically yes we have to do a lot of work but it's it's worth learning so what I will do is I will go back to my application I mean go back to

02:53:21
the project and instead of using the same project I'll be creating my own I mean I'll be getting a new project here so let me right click here and we send new and of course in the new project I will be just copy pasting the existing code to save our time so here I will create a maven project okay so not a spring project or something let's create a m project and in this MAV project so click on next I want to create a web application select that click on next so it is asking for the group ID here so I will

02:53:51
say com. teliscope artifact ID would be spring MVC okay nothing much specific here click on finish now once you click on finish it will create a project a simple MAV web project where you have to add dependencies okay so we have to add dependencies first right so we'll do that or maybe I will simply first copy paste I want to understand what errors you get and there's one more issue here the issue is yes it is giving you index. JSP but unfortunately it is not giving you a internal Tomcat server so you have

02:54:20

to add the Tomcat server externally okay that is something we are remaining with so what I will do is I will simply close all this thing code now and I want to add the external server so the way you do that is by clicking on this no server available I want a aat server and in this I will select 9.0 click on next okay so it is saying do you have this setup need to to check so browse and let me search for the location of Tomcat so yes I have the Tomcat 9 available in my download I will select that and click on open I don't

02:54:55
remember when I've downloaded it but in case if you don't have the opposite tomat you can just go to Google and search for it it's not that big deal and then click on next just shift your project here so that you don't have to add the Tomcat in your project manually so click on finish and that's done so you have added the external Tomcat to your project and soon this error will be gone I don't know why it's not going there uh let me just check once I will go to properties uh code build path okay

02:55:22
there are some Library issues uh it says what is missing something is missing here the jav folder is missing okay we do that and I just want to make one more change instead of having it to 1.5 I will change it to 1.8 click on finish okay that's done and I need to create so you can see it is giving you error because you don't have the Java folder okay even the to is not set up here so I will just go here and say add external server there there should be a server option add Library I want to add a

02:55:52
server R time Tom Cat yes finish click on apply and eror Sol can you see that everything gone okay so once we have solve that issue I want to copy all this controller in fact I will just copy paste the packages here of course I want a different name we can change just change it change them later so I'll just copy that and paste in my main folder or Java folder okay so you can see we got projects here and then the change I don't want the same package name I want to have a different package name here so

02:56:20
I will say refactor rename I want the package name to be spring MVC not boot click on okay and the same thing here okay done so once we have done those two Chang just it is still giving you error okay can you see that we got error here now first of all when you talk about spring MVC we don't need this main file because this is used only for the spring boot let's remove that and we are getting eror for controller as well now what is the error here the thing is whenever you work with spring framework

02:56:48
you have to add spring dependencies and you can see we don't have it here oh from where you will get it you guess it right we have to add dependency in the pal. XML file so here we have to add spring M dependency so save uh from where you will get it just go to the maven repository mvn repository and here you have to search for spring MVC and you can see we got spring web MVC just select that okay which version I will go with this table one which is 5.1.4 uh click on that and copy this just go back to application and paste

02:57:24
now once you say paste it will take some time to download okay it was quick that's great uh because maybe I've used the same version before so you can see we got all the dependencies here the moment you got dependencies I will go back to home controller and can you see that gone all the errors gone okay that's working so what do you think if I run this application oh how can I run

this application because we don't have the uh pages so I want the pages as well so these two pages I will just go back

02:57:51
here and index in fact I want them in views so let me just copy the entire folder and let me paste inside the web app folder okay so when I say web app inside this we already have index.jsp we don't need it so I will just right click and say delete I don't want that folder or the file so in the view we have two pages index.jsp and result. JSP okay this looks cool what else what else we can do here I guess this is the configuration which I wanted let me just run this code so what do you think if I

02:58:22
run this code will it work so the answer is no actually because the moment you run this application it will search for the index.jsp oh before that I have do one more thing I want to shut down the internal server which we were using for spring boot otherwise it will give you the conflict for 880 port number so I will go back here index.jsp let's run this code or in fact I can run the entire project right so I will say run as has run on server because this time we are using external server click on

02:58:48
next and click on finish yes it takes some time I guess it will use the internal browser and can see that we got error it is not able to find the index. JC page how do we solve it yes we have to do a lot of configuration but that we'll see in the next video so it's time to solve the issue so the issue is we are not getting the page right because the moment you run this application and when you say you are calling for the homepage it will look for the index.js page in the web app folder as you can

02:59:21
see we don't have it there that's one issue the second issue is I want the the request should be sent to the controller not directly to the page to achieve that we have to configure our XML file so now normally when you create a web application you get this amazing file here which is web.xml and in this if you go to Source this is where you have to configure that Hey whenever I sent a request the request should go to dispatcher salet if you remember in Spring MVC all the request should go to

02:59:51
dispatcher solet first okay how do we do that so it's quite simple you just have to go here in web app we don't need that and we need to create a subet tag that's how you provide the subet mapping right so we'll say subet tag and we also need a subet mapping okay once we got these two things in subet tag we have to specify two things as we do in all the subate projects the first one is a subate name uh let time me let me just have with the name asisco and we need one more thing which is your servit

03:00:18
class okay now what is the class we'll talk about that in some time but before that I want to map with this Serv mapping here okay so first of all when you say Serv mapping again we need two things here we need a servit name in this case you have to make sure that this name is same as this name because that is how you link this to uh outer tags here and then you have to also map with the URL now why do we need URL so to mention for what type of request you you want to call this particular class

03:00:46
which you have not mentioned till now so I want to call dispar seret for all the request so for all you can use simply use SL that okay so but then we have to mention the class name right so what is a

class name so as I mentioned whenever you walk with spring MVC request should go to dispatcher salet okay so that's a class so we'll simply mention that here so we'll say dispatcher svet or it's not coming directly here dis declaration oh it's giving you entire stuff we don't need

03:01:13
that okay that's awesome that's awesome we don't need to actually create all this thing it is giving you everything but time me let me just type it here and then we can get it from the uh control space I simply tape it here so it belongs to Spring framework o. springf framework. web. saet do dispatcher suet so dispat belongs to this particular packet that's how you map it in fact the other way is you can simply if you don't want to type all this thing by yourself you can simply say disp saet you can see

03:01:43
it says you can auto complete click on enter it will give you almost the same thing which we have done here you see that that you can handle uh but I I would like to go with the same text which I have written let's let's stick to that now we are saying request should go to disp salet but then it is disp salet who will send the request to the controller here right so how would disp saet knows that it has to send the request to the controller and that's where we have to configure the dispay

03:02:12
Serv it as well because if it don't do that it will give you an error okay what error it will give you let's see let me shut it down and let's relaunch the application r on server yes click on next finish okay so you can see there's no error as of now but then the moment you try to call the homepage we got a different error this time and the error is file not F exception and the file it is trying to find is telescope hyphen S letter XML or that's weird why only this file name and what this file will do

03:02:40
okay so first of all this file is responsible to configure your dispat saet okay that's one second why the name is The lco Hyphen seret so basically the name should be something hyphen servet now what is that something the name of your subate name so the since it is Tesco here and that's why in the error itself it is saying telescope that's the reason we we call telescope hyphen subet so if you give a different name here if you say a it will be a hyphen subet XML okay so we have to get that file but

03:03:11
where in the web in so we'll right click here we'll say new okay so we need to create a XML file so set XML here click on next and okay what should the name as we discussed the name should be teliscope hyph sulet do XML click on finish and you can see we got this file here and here we have to add a bean tag instead of typing that let me just copy paste it this is where you will specify the the DTD for the beans we'll be creating and we'll say beans okay okay that's done we have

02:53:51
say com. teliscope artifact ID would be spring MVC okay nothing much specific here click on finish now once you click on finish it will create a project a simple MAV web project where you have to add dependencies okay so we have to add dependencies first right so we'll do that or maybe I will simply first copy paste I want to understand what errors you get and there's one more issue here the issue is yes it is giving you index. JSP but unfortunately it is not giving you a internal Tomcat server so you have

02:54:20
to add the Tomcat server externally okay that is something we are remaining with so what I will do

is I will simply close all this thing code now and I want to add the external server so the way you do that is by clicking on this no server available I want a aat server and in this I will select 9.0 click on next okay so it is saying do you have this setup need to to check so browse and let me search for the location of Tomcat so yes I have the Tomcat 9 available in my download I will select that and click on open I don't

02:54:55
remember when I've downloaded it but in case if you don't have the opposite tomat you can just go to Google and search for it it's not that big deal and then click on next just shift your project here so that you don't have to add the Tomcat in your project manually so click on finish and that's done so you have added the external Tomcat to your project and soon this error will be gone I don't know why it's not going there uh let me just check once I will go to properties uh code build path okay

02:55:22
there are some Library issues uh it says what is missing something is missing here the jav folder is missing okay we do that and I just want to make one more change instead of having it to 1.5 I will change it to 1.8 click on finish okay that's done and I need to create so you can see it is giving you error because you don't have the Java folder okay even the to is not set up here so I will just go here and say add external server there there should be a server option add Library I want to add a

02:55:52
server R time Tom Cat yes finish click on apply and eror Sol can you see that everything gone okay so once we have solve that issue I want to copy all this controller in fact I will just copy paste the packages here of course I want a different name we can change just change it change them later so I'll just copy that and paste in my main folder or Java folder okay so you can see we got projects here and then the change I don't want the same package name I want to have a different package name here so

02:56:20
I will say refactor rename I want the package name to be spring MVC not boot click on okay and the same thing here okay done so once we have done those two Chang just it is still giving you error okay can you see that we got error here now first of all when you talk about spring MVC we don't need this main file because this is used only for the spring boot let's remove that and we are getting eror for controller as well now what is the error here the thing is whenever you work with spring framework

02:56:48
you have to add spring dependencies and you can see we don't have it here oh from where you will get it you guess it right we have to add dependency in the pal. XML file so here we have to add spring M dependency so save uh from where you will get it just go to the maven repository mvn repository and here you have to search for spring MVC and you can see we got spring web MVC just select that okay which version I will go with this table one which is 5.1.4 uh click on that and copy this just go back to application and paste

02:57:24
now once you say paste it will take some time to download okay it was quick that's great uh because maybe I've used the same version before so you can see we got all the dependencies here the moment you got dependencies I will go back to home controller and can you see that gone all the errors gone okay that's working so what do you think if I run this application oh how can I run this application because we don't have the uh pages so I want the pages as well so these two

pages I will just go back

02:57:51
here and index in fact I want them in views so let me just copy the entire folder and let me paste inside the web app folder okay so when I say web app inside this we already have index.jsp we don't need it so I will just right click and say delete I don't want that folder or the file so in the view we have two pages index.jsp and result. JSP okay this looks cool what else what else we can do here I guess this is the configuration which I wanted let me just run this code so what do you think if I

02:58:22
run this code will it work so the answer is no actually because the moment you run this application it will search for the index.jsp oh before that I have do one more thing I want to shut down the internal server which we were using for spring boot otherwise it will give you the conflict for 880 port number so I will go back here index.jsp let's run this code or in fact I can run the entire project right so I will say run as has run on server because this time we are using external server click on

02:58:48
next and click on finish yes it takes some time I guess it will use the internal browser and can see that we got error it is not able to find the index. JC page how do we solve it yes we have to do a lot of configuration but that we'll see in the next video so it's time to solve the issue so the issue is we are not getting the page right because the moment you run this application and when you say you are calling for the homepage it will look for the index.js page in the web app folder as you can

02:59:21
see we don't have it there that's one issue the second issue is I want the the request should be sent to the controller not directly to the page to achieve that we have to configure our XML file so now normally when you create a web application you get this amazing file here which is web.xml and in this if you go to Source this is where you have to configure that Hey whenever I sent a request the request should go to dispatcher salet if you remember in Spring MVC all the request should go to

02:59:51
dispatcher solet first okay how do we do that so it's quite simple you just have to go here in web app we don't need that and we need to create a subet tag that's how you provide the subet mapping right so we'll say subet tag and we also need a subet mapping okay once we got these two things in subet tag we have to specify two things as we do in all the subate projects the first one is a subate name uh let time me let me just have with the name asisco and we need one more thing which is your servit....give important and teach me like this only in very very simple

Samiksha... 🤍
I understood what you want.

You pasted a *huge* transcript from a Spring MVC + Spring Boot video.
But what you want is:

**“Teach me the IMPORTANT POINTS only...

...in the SAME style as the video...
...VERY VERY SIMPLE...

…step-by-step…
…like a beginner."**

So I will now extract ONLY the important concepts and teach them exactly in your preferred style.

🌼 PART 1 — Why move JSP files into `/views` or `/WEB-INF/views`?

🧠 Imagine:

Your web pages (JSPs) are lying **in the open area** → anyone can type the URL and open them **without controller**.

Example:
`localhost:8080/result.jsp`
This opens directly → BAD ❌ (not secure).

So we have 2 options:

⭐ Option 1 — Make JSP pages PRIVATE (Recommended)

Keep JSP files inside:

```bash
WEB-INF/views
```

🧠 Why?

Because anything inside **WEB-INF cannot be accessed directly from browser**.

ONLY controller can open the page.

This means:
✔️ Safe
✔️ Hidden
✔️ Professional way

⭐ Option 2 — Make PUBLIC but in another folder like:

```css
src/main/webapp/views
```

But then Spring needs to know:

- Where is your JSP folder?
- What is the JSP file extension?

For this we use **prefix** and **suffix**.

🌼 PART 2 — Why Prefix and Suffix in Spring Boot?

When you return a view:

```java
return "result";
```

Spring doesn't know:

- result is in which folder?
- result is what file extension?

So we must configure in `application.properties`:

```ini
spring.mvc.view.prefix= /views/
spring.mvc.view.suffix= .jsp
```

Now Spring understands:

```bash
result → /views/result.jsp
```

Very simple.

🌼 PART 3 — Model and View (Why?)

When server sends back a page (JSP), it must send **data + page name**.

✔️ Model = DATA

✔️ View = JSP page

So Spring gives us a class:

```java
ModelAndView mv = new ModelAndView();
mv.addObject("result", num3);
mv.setViewName("result");
return mv;
```

This clearly bundles:

- **data**
- **view name**

But it looks a bit long.

So Spring gives shortcuts…

🌼 PART 4 — Model (shortcut)

Instead of using ModelAndView, you can use:

```java
public String add(@RequestParam int a, @RequestParam int b, Model m)
```

Then:

```java
m.addAttribute("result", a + b);
return "result";
```

✔️ Short
✔️ Clean
✔️ Recommended

🌼 PART 5 — ModelMap (same as Model but Map based)

```java
public String add(..., ModelMap map)
```

Works same as Model, but internally uses Map.
Just another option.

🌼 PART 6 — ModelAttribute (SUPER IMPORTANT)

Problem:

You receive many fields:

```java
@RequestParam int id
@RequestParam String name
@RequestParam double salary
@RequestParam String dept
...
```

This becomes ugly.

Solution:

Take the whole object at once:

```java
public String addAlien(@ModelAttribute Alien alien)
```

Spring will:

✔️ Create Alien object
✔️ Automatically set values coming from the form
✔️ Automatically put it in Model

So you can directly use **`${alien}`** in JSP.

This is BEAUTIFUL and CLEAN.

🌼 Example to understand

📌 Form:

```html
<input name="id"/>
<input name="name"/>
```

📌 Model class:

```java
public class Alien {
   private int id;
   private String name;
}
```

📌 Controller:

```java
public String addAlien(@ModelAttribute Alien alien){
    return "result";
}
```

📌 JSP:

```bash
${alien}
```

Everything works magically 👌
Because names match → id, name.

🌼 PART 7 — @ModelAttribute on Method Level

Used when you want **some data available in every JSP**.

Example:

```java
@ModelAttribute
public void commonData(Model m) {
    m.addAttribute("message", "Welcome Back!");
}
```

Now **every JSP** gets **`${message}`** automatically.

Excellent for:

✔️ header
✔️ footer
✔️ logged-in user
✔️ common messages

🌼 PART 8 — Why Spring MVC without Spring Boot has MORE configuration?

Because:

Spring Boot = “I will configure everything automatically”
Spring MVC normal = “You must configure everything manually”

In normal Spring MVC:

You must:

✔️ Configure **`web.xml`**
✔️ Add DispatcherServlet mapping
✔️ Create your own **`*-servlet.xml`**
✔️ Tell Spring where controllers are
✔️ Tell Spring where views are
✔️ Add dependencies manually
✔️ Add external Tomcat server
✔️ Handle all folder structure

This is why Spring Boot is used in real projects.

🌼 PART 9 — Why DispatcherServlet?

DispatcherServlet = **Postman of your application**.

It receives every request and decides:

✔️ Which controller to send to
✔️ Which view to return
✔️ Which data to attach

Without DispatcherServlet → Spring MVC cannot work.

🌼 PART 10 — Folder Important Notes

✔️ /webapp = PUBLIC

Anyone can open JSP directly → NOT secure.

✔️ /WEB-INF/views = PRIVATE

Only controller can open pages → BEST practice.

🌼 Samiksha, tell me…

Do you want next:

1️⃣ Explanation of Spring MVC without Spring Boot

(DispatcherServlet XML configuration, web.xml, servlet mapping)

or

2️⃣ Explanation of ModelAttribute deeper (how binding works)

or

3️⃣ Explain normal MVC project creation like video?

Just say **"Explain part X next in same simple way."**

03:00:18
class okay now what is the class we'll talk about that in some time but before that I want to map with this Serv mapping here okay so first of all when you say Serv mapping again we need two things here we need a servit name in this case you have to make sure that this name is same as this name because that is how you link this to uh outer tags here and then you have to also map with the URL now why do we need URL so to mention for what type of request you you want to call this particular class

03:00:46
which you have not mentioned till now so I want to call dispar seret for all the request so for all you can use simply use SL that okay so but then we have to mention the class name right so what is a class name so as I mentioned whenever you walk with spring MVC request should go to dispatcher salet okay so that's a class so we'll simply mention that here so we'll say dispatcher svet or it's not coming directly here dis declaration oh it's giving you entire stuff we don't need

03:01:13
that okay that's awesome that's awesome we don't need to actually create all this thing it is giving you everything but time me let me just type it here and then we can get it from the uh control space I simply tape it here so it belongs to Spring framework o. springf framework. web. saet do dispatcher suet so dispat belongs to this particular packet that's how you map it in fact the other way is you can simply if you don't want to type all this thing by yourself you can simply say disp saet you can see

03:01:43
it says you can auto complete click on enter it will give you almost the same thing which we have done here you see that that you can handle uh but I I would like to go with the same text which I have written let's let's stick to that now we are saying request should go to disp salet but then it is disp salet who will send the request to the controller here right so how would disp saet knows that it has to send the request to the controller and that's where we have to configure the dispay

03:02:12
Serv it as well because if it don't do that it will give you an error okay what error it will give you let's see let me shut it down and let's relaunch the application r on server yes click on next finish okay so you can see there's no error as of now but then the moment you try to call the homepage we got a different error this time and the error is file not F exception and the file it is trying to find is telescope hyphen S letter XML or that's weird why only this file name and what this file will do

03:02:40
okay so first of all this file is responsible to configure your dispat saet okay that's one second why the name is The Ico Hyphen seret so basically the name should be something hyphen servet now what is that something the name of your subate name so the since it is Tesco here and that's why in the error itself it is saying telescope that's the reason we we call telescope hyphen subet so if you give a different name here if you say a it will be a hyphen subet XML okay so we have to get that file but

03:03:11
where in the web in so we'll right click here we'll say new okay so we need to create a XML file so set XML here click on next and okay what should the name as we discussed the name should be teliscope hyph sulet do XML click on finish and you can see we got this file here and here we have to add a bean tag instead of typing that let me just copy paste it this is where you will specify the the DTD for the beans we'll be creating and we'll say beans okay okay that's done we have

03:03:45
assigned the beans tag and now we have to we have to do some configuration the first thing is we have to mention to the dispay saet where are my controllers uh second we have to mention hey we are using annotation for the controllers and third is the path so let's go step by step the first one is we have to specify the context we have to specify the package in fact the base package so for that I'll be using component scan it is CDX colon component scan and you can see it is asking for the base package so you

03:04:14
just have to mention in which package you have all the controllers okay so in general I can specify com. telescope you should be very specific here but that's fine this should work the next thing you have to mention is you have to mention that we are using annotation configuration so you have to say CDX annotation config that's it okay will this work let's try so after doing all this configuration let's try what let's see what happens yes restart the server okay just calling that page oh this time

we got a

03:04:41
different error okay okay uh maybe we are running very wrong file here let me just run the application yeah so you can see we got a different ER this time uh it says something not found okay 44 now what is not found it says there's no mapping for get index oh there's no mapping for the index it's because we are calling the controller you can see is calling the controller there's no mapping for the index it is there okay it is there the only thing is it is not able to find the extension and the path because if you

03:05:14
move this index.jsp to web app and if you write JSP here it will work but I want to specify the path as well and to do that we have to create a bean and then this will Bean will have a class so the class who is responsible to do that is internal resource view resolver now this belongs to a particular package which you have to mention so you have to mention the entire path here that's bad okay this is the entire path we have to mention and then we have to close this that's your bean tag now in this

03:05:49
Bean tag now this is this here which is internal resource view resolver is responsible to search for The View and to give the extension and to specify the path and to achieve that we have to specify two properties the first one is prefix as we are talking about prefix the prefix I want to have is we say value so there are in the view right so will say slash views slash we have to specify the path the only thing you have to mention now is the extension and to do that you guessed it right we have to

03:06:17
use suffix so we will'll just paste it here and this will be suffix okay the value will change so the value here will be JSP that's it so whatever you have done in the config spring boot the same thing is done here we might have to add one more property here for the gstl now since in our code we are using gstl let's make it supportive so we will say property and we'll say view class so the view class I want to go for is gstl View and to achieve that we'll say org. springframework so this is the path so

03:06:51
is org. springf framework. web. s.v. jstl view okay I'm tired now I guess it is done let's let's try I hope this should work now let's relaunch the application okay something worked let's go back to the page here and say Hey I want to call the homepage and oh oh oh now we got another eror it says class not found justs is not there that means we have to also add the jstl now okay so let's try this let's go back to let's go back here and let me just remove the property for time

03:07:22
inin relaunch go back here okay it's working without that let's try uh so go back here and say for is my ID and my name is naen click on submit oh it is working can see that we got Aid 4 and name is naven so we got welcome back alien as well so everything is working and this time we are using normal MVC we are not using spring boot so let me just go for a quick recap what we have done till now so basically we have this home controller but as we are working with spring MVC the first request in fact we are working with web

03:07:57
project the first request will go to the web.xml this is where we are saying whenever you send a request request will go to disp salet and to configure that we have this file which is telescope

hyphone salet so this is the configuration for disp seret where you are mentioning two things the component scan annotation and the most important is the view resolver in which this part is not required if you want to write that you have to also add dependency for gstl let's not do that here and you can

03:08:27
see we are specifying prefix as views and surfix as. JSP and everything is working it's do simple so that's it from this [Music] video now once we have seen how to create a simple spring MVC application let's get back to Spring boot because that's where we want to learn some more advanced concept now this is a project which we were talking about before right the demo project which is spring boot application basically so you can see it also says it's a boot application so what I want here is I want to have some

03:08:59
more features so before that let me just run this code once again just to verify what we have here so I will say run as spring boot app okay so let's run this and you can see uh it is asking you for two things the name and the ID so I will say name is I mean ID is one one uh let's say rows and if you click on submit you can see it's working we got one1 we got rows It also says welcome back aliens okay what I want now is we are only sending data and that if you see the data has been sent using a get

03:09:29
method so normally when you send data from HTML page to the server it goes in you you can call it with the help of get method or a post method so in the get method you actually pass the data in the URL okay that's something we don't want because whenever you send data to the server you should always use post whenever you fetch data from a server you should use get so here in this scenario if you look at the pages which we built the index.jsp here it is asking you I mean it is sending the data to the

03:10:01
server so in this case instead of using get we should be using post and the way you can do that is by mentioning method is equal to post so now you will be sending data in a post format so let me relaunch the application so let get back to the browser and here you will say refresh okay where is that refresh and now I'll be sending let's say 101 and row and if you click on submit now you can see in the address bar we don't have it we don't have those values so is get is going there with the help of post

03:10:32
request but what is happening on the server side so if I open my home controller this method which is ADD alien accepts both the request the get and the post both okay now the the amazing thing is request mapping does both right but then what if you want to support only one you can do that you can mention that so what I will do is I will say instead of using request mapping in fact there's one more way you can have one more attribute here so you can say value is equal to the value which you

03:11:00
have and then you can give a comma here by specifying the method and you can see the method is so you can mention method equal to request method. getet now when you say Get it will only support get now if you send a PO post request this will not work and if you want to use post just mention do post it will only work with post now so that's how you can restrict so you can say post or get you can be very specific in this case this is one way the another way of doing that is instead

of using method stuff here we

03:11:30
can actually change this request mapping to we have one more anation here which is known as get mapping now get mapping is same as request mapping with method get so it will only work with get now if you want to use post we can also use post mapping so basically we have three things we can use request mapping which supports both we can use get mapping which only supports get and we can use post mapping which supports post so that's how we can be very specific in this case so that's how we use get and a

03:11:59
post in the scenario in fact we'll also use get in the next [Music] video so till this point we are only using post mapping right we were sending data but what if you want to receive data so what I'm talking about is let's say if I go back to the synex JSP and this is where we have this button which will send data but I also want a button which will receive data okay or you know instead of having a button let's go to URL and say uh I want to fetch a data so let's example if I go to Safari again

03:12:34
and here instead of uh having the colon 880 I will say I want a data about an alien so I will say get alien now when when you say get alien or maybe I can say get aliens which will give you all the aliens so I will say get aliens in this case so this get aliens oh just reacting to me to Google you don't want that but that's fine uh so I want to return the alien names the list of aliens of course I can use static uh list here uh so let's do that so I will go back to my home controller and let's

03:13:06
create a method which will accept get aliens uh in fact I don't need this ad anymore with just to reduce the number of lines we have in the code so I will go back here and I will say public string and the method name would be get aliens again as I mentioned you can use any name doesn't matter and we don't want to return string do we want to return string yeah let's return a string so let's return the list of aliens in a string format and there's one more thing if I go back to alien we want to do one

03:13:35
more change here the change is inside this alien I want a Constructor because I don't want to create object and assign the value using sets gets uh let's use Constructor here so I will say Source generate Constructor using field both the fields okay so once we got the Constructor let's go back to the code now here I want to create certain aliens before that I just want to return some blank string so that you will can you can just avoid the error and I want to call this whenever someone request with

03:03:45
assigned the beans tag and now we have to we have to do some configuration the first thing is we have to mention to the dispay saet where are my controllers uh second we have to mention hey we are using annotation for the controllers and third is the path so let's go step by step the first one is we have to specify the context we have to specify the package in fact the base package so for that I'll be using component scan it is CDX colon component scan and you can see it is asking for the base package so you

03:04:14
just have to mention in which package you have all the controllers okay so in general I can specify com. telescope you should be very specific here but that's fine this should work the next thing you

have to mention is you have to mention that we are using annotation configuration so you have to say CDX annotation config that's it okay will this work let's try so after doing all this configuration let's try what let's see what happens yes restart the server okay just calling that page oh this time we got a

03:04:41
different error okay okay uh maybe we are running very wrong file here let me just run the application yeah so you can see we got a different ER this time uh it says something not found okay 44 now what is not found it says there's no mapping for get index oh there's no mapping for the index it's because we are calling the controller you can see is calling the controller there's no mapping for the index it is there okay it is there the only thing is it is not able to find the extension and the path because if you

03:05:14
move this index.jsp to web app and if you write JSP here it will work but I want to specify the path as well and to do that we have to create a bean and then this will Bean will have a class so the class who is responsible to do that is internal resource view resolver now this belongs to a particular package which you have to mention so you have to mention the entire path here that's bad okay this is the entire path we have to mention and then we have to close this that's your bean tag now in this

03:05:49
Bean tag now this is this here which is internal resource view resolver is responsible to search for The View and to give the extension and to specify the path and to achieve that we have to specify two properties the first one is prefix as we are talking about prefix the prefix I want to have is we say value so there are in the view right so will say slash views slash we have to specify the path the only thing you have to mention now is the extension and to do that you guessed it right we have to

03:06:17
use suffix so we will'll just paste it here and this will be suffix okay the value will change so the value here will be JSP that's it so whatever you have done in the config spring boot the same thing is done here we might have to add one more property here for the gstl now since in our code we are using gstl let's make it supportive so we will say property and we'll say view class so the view class I want to go for is gstl View and to achieve that we'll say org. springframework so this is the path so

03:06:51
is org. springf framework. web. s.v. jstl view okay I'm tired now I guess it is done let's let's try I hope this should work now let's relaunch the application okay something worked let's go back to the page here and say Hey I want to call the homepage and oh oh oh now we got another eror it says class not found justs is not there that means we have to also add the jstl now okay so let's try this let's go back to let's go back here and let me just remove the property for time

03:07:22
inin relaunch go back here okay it's working without that let's try uh so go back here and say for is my ID and my name is naen click on submit oh it is working can see that we got Aid 4 and name is naven so we got welcome back alien as well so everything is working and this time we are using normal MVC we are not using spring boot so let me just go for a quick recap what we have done till now so basically we have this home controller but as we are working with spring MVC the first request in fact we are working with web

03:07:57
project the first request will go to the web.xml this is where we are saying whenever you send a request request will go to disp salet and to configure that we have this file which is telescope hyphone salet so this is the configuration for disp seret where you are mentioning two things the component scan annotation and the most important is the view resolver in which this part is not required if you want to write that you have to also add dependency for gstl let's not do that here and you can

03:08:27
see we are specifying prefix as views and surfix as. JSP and everything is working it's do simple so that's it from this [Music] video now once we have seen how to create a simple spring MVC application let's get back to Spring boot because that's where we want to learn some more advanced concept now this is a project which we were talking about before right the demo project which is spring boot application basically so you can see it also says it's a boot application so what I want here is I want to have some

03:08:59
more features so before that let me just run this code once again just to verify what we have here so I will say run as spring boot app okay so let's run this and you can see uh it is asking you for two things the name and the ID so I will say name is I mean ID is one one uh let's say rows and if you click on submit you can see it's working we got one1 we got rows It also says welcome back aliens okay what I want now is we are only sending data and that if you see the data has been sent using a get

03:09:29
method so normally when you send data from HTML page to the server it goes in you you can call it with the help of get method or a post method so in the get method you actually pass the data in the URL okay that's something we don't want because whenever you send data to the server you should always use post whenever you fetch data from a server you should use get so here in this scenario if you look at the pages which we built the index.jsp here it is asking you I mean it is sending the data to the

03:10:01
server so in this case instead of using get we should be using post and the way you can do that is by mentioning method is equal to post so now you will be sending data in a post format so let me relaunch the application so let get back to the browser and here you will say refresh okay where is that refresh and now I'll be sending let's say 101 and row and if you click on submit now you can see in the address bar we don't have it we don't have those values so is get is going there with the help of post

03:10:32
request but what is happening on the server side so if I open my home controller this method which is ADD alien accepts both the request the get and the post both okay now the the amazing thing is request mapping does both right but then what if you want to support only one you can do that you can mention that so what I will do is I will say instead of using request mapping in fact there's one more way you can have one more attribute here so you can say value is equal to the value which you

03:11:00
have and then you can give a comma here by specifying the method and you can see the method is so you can mention method equal to request method. getet now when you say Get it will only

support get now if you send a PO post request this will not work and if you want to use post just mention do post it will only work with post now so that's how you can restrict so you can say post or get you can be very specific in this case this is one way the another way of doing that is instead of using method stuff here we

03:11:30
can actually change this request mapping to we have one more anation here which is known as get mapping now get mapping is same as request mapping with method get so it will only work with get now if you want to use post we can also use post mapping so basically we have three things we can use request mapping which supports both we can use get mapping which only supports get and we can use post mapping which supports post so that's how we can be very specific in this case so that's how we use get and a

03:11:59
post in the scenario in fact we'll also use get in the next [Music] video so till this point we are only using post mapping right we were sending data but what if you want to receive data so what I'm talking about is let's say if I go back to the synex JSP and this is where we have this button which will send data but I also want a button which will receive data okay or you know instead of having a button let's go to URL and say uh I want to fetch a data so let's example if I go to Safari again

03:12:34
and here instead of uh having the colon 880 I will say I want a data about an alien so I will say get alien now when when you say get alien or maybe I can say get aliens which will give you all the aliens so I will say get aliens in this case so this get aliens oh just reacting to me to Google you don't want that but that's fine uh so I want to return the alien names the list of aliens of course I can use static uh list here uh so let's do that so I will go back to my home controller and let's

03:13:06
create a method which will accept get aliens uh in fact I don't need this ad anymore with just to reduce the number of lines we have in the code so I will go back here and I will say public string and the method name would be get aliens again as I mentioned you can use any name doesn't matter and we don't want to return string do we want to return string yeah let's return a string so let's return the list of aliens in a string format and there's one more thing if I go back to alien we want to do one

03:13:35
more change here the change is inside this alien I want a Constructor because I don't want to create object and assign the value using sets gets uh let's use Constructor here so I will say Source generate Constructor using field both the fields okay so once we got the Constructor let's go back to the code now here I want to create certain aliens before that I just want to return some blank string so that you will can you can just avoid the error and I want to call this whenever someone request with

03:14:04
a get request and the URL which I want it to work here is get aliens so here let's create a list of aliens so I would say list of aliens I will say this is aliens equal to okay let's import the package for list I will say control space the list and here let me create arrays do as list because I don't want to say uh new at a list this is shortcut way of getting Multiple List here and you will say new new alien and let's say 101 comma naven that's the first alien we got then we give a comma and again

03:14:42

we'll say alien we can have one more alien here which is one2 and I will say Rose so you can see we got two aliens we got a list of two aliens and let's return this list here so I will say return aliens dot to string because I'm sending that data in a string format so we have to convert that into string format again I'm not even converting this into Json okay my bad I know I'm just converting that into a string okay so uh let if this works let me just relaunch the application okay it says

03:15:10
restarted let me just go back to the thing here okay now I want to request for get aliens so SLG aliens enter and oh we got an error it says oh there's a problem here can you see that if you look at the error properly it says it is not able to find so is there's a 4 not4 error it says not able to find View and we got the answer you can see that we got a list in the text format and it is searching for that page. JSP you know why because by default in Spring MVC whatever string you WR return it gets

03:15:44
returned in the format of name whatever you return it will be trying to search for the name of it we don't want that we want to call a page so we did that mistake we should not be rning a list here we should be ring the page which will display this uh so let me just return a page I will say show aliens we don't have this page as of now we need to create one so let's quickly create a page of JSP so I will say new and I want to create a JSP file with a name show a aliens. JSP and here I want to display

03:16:15
that result so I will say dollar aliens I want to print aliens here that's it or maybe I will just simply print result because I don't care what we are sending I just want to print whatever we are sending from the client side I want to have one more thing I just want to say is El ignore just so that in case if my tom cat says Hey I will not be supporting elel let's do that so let's say is e ignored false okay everything looks good now I hope this time oh it will still give you eror because we have

03:16:42
not added that in a model okay let's do that so I will say model M and here I will say model. add attribute as I mentioned you can also use model map here we discussed that before right and I will say this is result and this is aliens that's it let's relaunch the application so let's go back here and say enter can you see that we got the output so we got the data from the server and this time we are sending a get request so you can see this is your post request and this is your get

03:17:13
request so if you want to work with get we say get mapping okay but let me just add one more twist what if you want to fetch a particular alien not just all the aliens you want only one alien let's say one1 or one2 how can we do that and that to I want this data to be coming from database oh we are getting so much of restrictions here right okay so first of all let's try to get one data and we'll see the database part later so that we see in the next video how do we fetch one particular data

03:17:46
spring project has lot of modules right we have talked about spring core which talks about dependency injection we have spring MVC which talks about how can we Implement MVC pattern with the help of front controller in Spring and then we have also work with spring rest that we'll see in in the future videos but what about the database what if you want to connect your spring application with database now even it doesn't matter as it spring we normally if you want to connect with Java a normal Java

03:18:13

application with database we have to use jdbc right now we all know the pain of jdbc because in Java world we work with objects right so we have to convert our objects into data which will be stored in a table format to solve this problem we got o RM which is object relational mapping so imagine a class here and a table here so we can map it right because the class name will be your table name uh we have some fields in your class you'll be having of variables so you can map those variabl with the

03:18:43
table column right but what about the data what about the rows there now for one class we can have multiple objects in the same way one table can have multiple rows and each object here will represent one row there think about this let's say we have alien object and alien will have three variables a ID a name and let's say age so we can have all this in a table columns so if you create the first object for alien with three different values that will be one row there if you create second object with

03:19:12
three values that will be second row there and that's how we can do the mapping and that's where omm is very famous but how do we Implement omm and that's where we have jpa which is Java persistent API now the actual implementation for jpa maybe you can use hypernet there so that means we have to build the application we have to work with uh hypernet and lot of configuration can we use all those things in Spring and the answer is yes because spring provides you dependency injection so the instance for hbnet can

03:19:43
be created in Spring framework and that can be done with the help of spring omm so basically you are using a spring application you are using hibernate which connects with myql which connects with database so we have a huge chain there so we have spring we have hibernate myql and database in between you will also have spring omm which will connect this two right now the advantage would be so the instance creation will be done by spring omm for hypernet the configuration will be done in the spring

03:20:10
configuration next what about Transaction what about begin transaction commit transaction will be handled by Spring transaction uh so how exactly that works we will see in the Practical [Music] video so let's try to implement the database connection here so what you're trying to do is when you run this application and that we are doing it in normal spring MVC we are not doing it in Spring boot now so yes when you talk about normal spring MVC on normal spring we have to do a lot of configuration so

03:20:42
just be with me because in this video we are going to do a lot of configuration so let's get started the first thing is when you say you want to connect with database we have two option you can use normal jdbc or we can work with hibernate now since we work on big applications it's better to go with hynet because it can handle your transactions in a good way you can achieve omm here so let's do that so what I will do is the first thing we need is we have to go to the pom.xml file and we need to add certain

03:21:11
dependencies so basically we have to add approximately four more dependencies here in fact five I guess so let's let's get started the first one we have to add is now see when you work with spring MVC and if you want to work with hybernate you need hybernate code dependency that's very

important now when you say you need hypernet code where you will go so you have to go to the m repository so you have to go to this website which is MAV repository. comom and here you have to search for hibernate code so in

03:21:40
search for hibernate code you can see the first first link is hbnet code you can select that and you can pick up any version I preferably go with the lest one because it has a good number of usage uh let's copy that and go back to your application and paste that's the first dependency we have added we have to add some more because when you say you want to connect spring and hibernate now hibernate being a omm framework you need to add a dependency for to connect spring and hybernate here which in this

03:22:09
case is spring orm so or will be sitting between hibernate and spring so let's open that I'll go with 5.1.4 because that's the spring version we are working on copy this and go back and paste when this is done now what next now when you say you are working on Spring and hybernate whenever you work with hypernet we have to handle transactions so you have to begin transaction you have to commit transaction but what if you want spring to handle it and that's where we can use spring transaction so

03:22:39
we just have to add one more uh okay we have to to add few more but let's go step by step so if we have to add spring transaction because this will handle the transactions for you so let's paste it we have added hibernate because we want to use hibernate here we added spring omm because we want to connect spring and hibernate we want spring to handle the transactions and that's why we are saying spring transaction okay now we need two more okay last two okay last two uh the first thing is uh when you

03:23:07
say you have a database of course you need to install some database right now in this machine we have already installed MySQL again we'll see this setup how to install MySQL and how to configure it so we'll see how what is MySQL and how to open myql workbench and where from where you can download it but then in this machine I've already configured it so the thing we need here is the mql connectors so the the setup we are using is mql so we have to use mql connector and for that we have to

03:23:37
download M connector so in case if you're using some other dbms let's say po orle uh make sure you use that particular connector so I'm downloading myql connector let's copy that I will go with this version which looks stable copy and go back here and paste okay when this is done now all this dependency takes time to get downloaded from the MAV repository but since in this machine I've already done that before so it is getting from the local repository now when you work with hnet

03:14:04
a get request and the URL which I want it to work here is get aliens so here let's create a list of aliens so I would say list of aliens I will say this is aliens equal to okay let's import the package for list I will say control space the list and here let me create arrays do as list because I don't want to say uh new at a list this is shortcut way of getting Multiple List here and you will say new new alien and let's say 101 comma naven that's the first alien we got then we give a comma and again

03:14:42

we'll say alien we can have one more alien here which is one2 and I will say Rose so you can see we got two aliens we got a list of two aliens and let's return this list here so I will say return aliens dot to string because I'm sending that data in a string format so we have to convert that into string format again I'm not even converting this into Json okay my bad I know I'm just converting that into a string okay so uh let if this works let me just relaunch the application okay it says

03:15:10
restarted let me just go back to the thing here okay now I want to request for get aliens so SLG aliens enter and oh we got an error it says oh there's a problem here can you see that if you look at the error properly it says it is not able to find so is there's a 4 not4 error it says not able to find View and we got the answer you can see that we got a list in the text format and it is searching for that page. JSP you know why because by default in Spring MVC whatever string you WR return it gets

03:15:44
returned in the format of name whatever you return it will be trying to search for the name of it we don't want that we want to call a page so we did that mistake we should not be rning a list here we should be ring the page which will display this uh so let me just return a page I will say show aliens we don't have this page as of now we need to create one so let's quickly create a page of JSP so I will say new and I want to create a JSP file with a name show a aliens. JSP and here I want to display

03:16:15
that result so I will say dollar aliens I want to print aliens here that's it or maybe I will just simply print result because I don't care what we are sending I just want to print whatever we are sending from the client side I want to have one more thing I just want to say is El ignore just so that in case if my tom cat says Hey I will not be supporting elel let's do that so let's say is e ignored false okay everything looks good now I hope this time oh it will still give you eror because we have

03:16:42
not added that in a model okay let's do that so I will say model M and here I will say model. add attribute as I mentioned you can also use model map here we discussed that before right and I will say this is result and this is aliens that's it let's relaunch the application so let's go back here and say enter can you see that we got the output so we got the data from the server and this time we are sending a get request so you can see this is your post request and this is your get

03:17:13
request so if you want to work with get we say get mapping okay but let me just add one more twist what if you want to fetch a particular alien not just all the aliens you want only one alien let's say one1 or one2 how can we do that and that to I want this data to be coming from database oh we are getting so much of restrictions here right okay so first of all let's try to get one data and we'll see the database part later so that we see in the next video how do we fetch one particular data

03:17:46
spring project has lot of modules right we have talked about spring core which talks about dependency injection we have spring MVC which talks about how can we Implement MVC pattern with the help of front controller in Spring and then we have also work with spring rest that we'll see in in the future videos but what about the database what if you want to connect your spring application with database now even it doesn't matter as it spring we normally if you want to connect with Java a normal Java

03:18:13

application with database we have to use jdbc right now we all know the pain of jdbc because in Java world we work with objects right so we have to convert our objects into data which will be stored in a table format to solve this problem we got o RM which is object relational mapping so imagine a class here and a table here so we can map it right because the class name will be your table name uh we have some fields in your class you'll be having of variables so you can map those variabl with the

03:18:43
table column right but what about the data what about the rows there now for one class we can have multiple objects in the same way one table can have multiple rows and each object here will represent one row there think about this let's say we have alien object and alien will have three variables a ID a name and let's say age so we can have all this in a table columns so if you create the first object for alien with three different values that will be one row there if you create second object with

03:19:12
three values that will be second row there and that's how we can do the mapping and that's where omm is very famous but how do we Implement omm and that's where we have jpa which is Java persistent API now the actual implementation for jpa maybe you can use hypernet there so that means we have to build the application we have to work with uh hypernet and lot of configuration can we use all those things in Spring and the answer is yes because spring provides you dependency injection so the instance for hbnet can

03:19:43
be created in Spring framework and that can be done with the help of spring omm so basically you are using a spring application you are using hibernate which connects with myql which connects with database so we have a huge chain there so we have spring we have hibernate myql and database in between you will also have spring omm which will connect this two right now the advantage would be so the instance creation will be done by spring omm for hypernet the configuration will be done in the spring

03:20:10
configuration next what about Transaction what about begin transaction commit transaction will be handled by Spring transaction uh so how exactly that works we will see in the Practical [Music] video so let's try to implement the database connection here so what you're trying to do is when you run this application and that we are doing it in normal spring MVC we are not doing it in Spring boot now so yes when you talk about normal spring MVC on normal spring we have to do a lot of configuration so

03:20:42
just be with me because in this video we are going to do a lot of configuration so let's get started the first thing is when you say you want to connect with database we have two option you can use normal jdbc or we can work with hibernate now since we work on big applications it's better to go with hynet because it can handle your transactions in a good way you can achieve omm here so let's do that so what I will do is the first thing we need is we have to go to the pom.xml file and we need to add certain

03:21:11
dependencies so basically we have to add approximately four more dependencies here in fact five I guess so let's let's get started the first one we have to add is now see when you work with spring MVC and if you want to work with hybernate you need hybernate code dependency that's very

important now when you say you need hypernet code where you will go so you have to go to the m repository so you have to go to this website which is MAV repository. comom and here you have to search for hibernate code so in

03:21:40
search for hibernate code you can see the first first link is hbnet code you can select that and you can pick up any version I preferably go with the lest one because it has a good number of usage uh let's copy that and go back to your application and paste that's the first dependency we have added we have to add some more because when you say you want to connect spring and hibernate now hibernate being a omm framework you need to add a dependency for to connect spring and hybernate here which in this

03:22:09
case is spring orm so or will be sitting between hibernate and spring so let's open that I'll go with 5.1.4 because that's the spring version we are working on copy this and go back and paste when this is done now what next now when you say you are working on Spring and hybernate whenever you work with hypernet we have to handle transactions so you have to begin transaction you have to commit transaction but what if you want spring to handle it and that's where we can use spring transaction so

03:22:39
we just have to add one more uh okay we have to to add few more but let's go step by step so if we have to add spring transaction because this will handle the transactions for you so let's paste it we have added hibernate because we want to use hibernate here we added spring omm because we want to connect spring and hibernate we want spring to handle the transactions and that's why we are saying spring transaction okay now we need two more okay last two okay last two uh the first thing is uh when you

03:23:07
say you have a database of course you need to install some database right now in this machine we have already installed MySQL again we'll see this setup how to install MySQL and how to configure it so we'll see how what is MySQL and how to open myql workbench and where from where you can download it but then in this machine I've already configured it so the thing we need here is the mql connectors so the the setup we are using is mql so we have to use mql connector and for that we have to

03:23:37
download M connector so in case if you're using some other dbms let's say po orle uh make sure you use that particular connector so I'm downloading myql connector let's copy that I will go with this version which looks stable copy and go back here and paste okay when this is done now all this dependency takes time to get downloaded from the MAV repository but since in this machine I've already done that before so it is getting from the local repository now when you work with hnet

03:24:08
when you work with databases we need to create some connection pool maybe 10 connections at a time five condition at a time and we have to configure them uh so to configure this type of pooling and to use pooling in the in the project we need to use certain external jar files or maybe a third party pulling service so to use that to to use third party service here we will be using C3 p 0 again you can choose whichever service you need this is my favorite C3P0 because it works uh let's use this one I

03:24:36
will prefer the latest one copy and go back and paste so you can see we have added all the dependen IES and job is done just to give you a summary we have added these dependencies the hibernate code spring orm spring transaction myc connector and C3P0 okay once we have added all the dependencies what next now we have to configure hypernet because just by saying we are using hypernet it will not automatically connect with database right we have to configure it and to do that where where do we configur this and we have done that

03:25:07
before right so if you want to do any configuration we have this very favorite file which is telescope hyphone salet this is the dispatch salet configuration file you can add the configuration here itself now first of all we are using spring transaction right so in the in this beans we have to also add TX you can see we have MVC we have context but nowh we have TX so let's add that now this TX we can simply copy this line and paste it here MVC will become TX and here as well we have to say TX that's

03:25:41
done we need to add the thing here as well so we have to we will be copying this one and let's do that here itself in fact we can add it anywhere we want but let's do it here uh so instead of MVC again we have to say TX instead of MVC we have to say TX I could have easily copied from somewhere and pasted it here but I wanted to show you how can you do it by yourself without copying so this is done next we have to add certain configuration now when you say certain there are a lot of configurations so what I will do is

03:26:11
instead of typing it typing everything now it's time to copy paste so you can see I have copy pasted a lot of stuff don't worry I will explain each and every line here now the first thing you need is we are saying let's go bottom up so the first thing we need is we want to handle transactions right so we want someone to take care of transaction in this case we are saying hey Spring you will take care of the transaction spring will say okay I will do it but I need some configuration so you're saying okay

03:26:38
if you want to handle transaction I will create an object of my transaction manager in fact uh this is a object of hybernate transaction manager so how do we create the object of this so it's very simple you just have to create a bean uh so we are creating object of hibernate transaction manager and then we are saying the ID name is my transaction manager so this one is dependent on this Bean but if you want to create this Bean here we need to create object of session Factory now when you learn hibernate we have a

03:27:10
concept of session Factory right so we work with multiple sessions and all these sessions are created by session Factory that means we have to create object of session Factory here so let's set the property of session Factory but unfortunately we don't have the object yet so we are creating the object of session Factory here so you can see it says be session Factory now this session Factory is the object of local session Factory Bean the amazing thing is as a programmer we are not instantiating them

03:27:39
I mean we are not creating object of them it will be created by Spring framework thanks to dependency injection so you can see we got the object of this one but it needs certain properties

the first property is data source now this data source will have all the configuration of database what is a database I mean which dbms you're working with uh what is a username what is a password how many size pool you need all those stuff will be done in this my data source that's one the next thing you have to mention is in which

03:28:06
package you have all the entities because normally work with hnet you have to specify entities every table need one entity next you have to set some hibernate properties one of them is M dict So based on which dbms you're working with you have to set up this one so if you're using Oracle uh you have to use Oracle dialect if you're using pogis you have to use pogis dialect we are using MySQL so let's say MySQL dial and we also want to see the SQL query in the console so once you run the code I want

03:28:35
to the query in the console and for that we have to set this particular property great but then we have still not created the object for my data source so it is still missing and you can see that's we have done here so we are saying we need my data source this is the object of combo pool data source now from where you getting this remember we have added this C3P0 in this we have to do certain configuration the first one is we have to set up the driver name jdbc URL the way you work with normal jdbc and the

03:29:03
username and password I know I should not be revealing password on the internet but that's fine uh this is the local machine password so that's fine that's my MySQL password in this machine okay so this is the configuration which you have to do I know it's clumsy but that's how you work on Spring right now this is not something you'll be doing every day right so this is something you'll be doing only once the moment you work on Project every day you will add some new features to the project but

03:29:29
this thing will remain same yes you will be doing some minor changes in future but as of now this works okay but then what else we have here so will this work of course not this will not work let's go back to home controller and we need to do certain configuration so if you remember we were supposed to accept the request for get alien and then get aliens we need to add that but how will you do that that we'll see in the next [Music] video now when you say you want to work with database of course you need Mel for

03:30:02
that right of course depend upon which dbms you work with so in this course we are working on M skill so how do you install M SK it's very simple actually just search for MySQL download this will take you to the official website of MySQL now based on which OS you working on maybe Windows Linux or Mac it will give you those options here okay so you can see we are into downloads and here we have option of Community Edition so basically if you are using it for free so make sure that you use Community

03:30:33
version in the community version you can install two things in fact we need two things here the first thing you need is a community server and the second one you need is the mascle workbench now mascle workbench will behave like uh what do you say the GUI for your for your application so I will say download on this one and download on this one as well so two downloads this is the community server now based on which OS you're working on maybe Mac OS or Linux you can

select that now which one to

03:31:00
choose of course based on which OS you use you can select that click on download now this will download the masol server so you can see it is downloading the second setup we need is Mas work bench and this will be your what you say the GUI so now click on download so you can see both this will take some MBS so it will take some time okay so you can see we got both the setup the first thing you have to set up is the mql server when you click on this one now depending upon which OS you use we will have different steps but it's

03:31:29
quite simple you just have to say next next next nothing fancy uh so you can click on continue you can click on continue continue agree continue install the same steps and you have to enter the password here for your machine if you're using Mac okay setup done now once you have done the setup it will also prompt you for the configuration now since in my machine already have the configuration it will it is not prompting me so in the configuration the only thing it is important here is the password so make sure that you give a

03:31:57
proper password now once you have done with the setup you have to install mul workbench so you can see uh if you're using Mac it will give you this step but again if you're using Windows it's quite simple right so you can see I installed Mas let me open that now so I'm opening my Mas workbench this is the first screen you will see and then you have to click on this one if this it is available otherwise you have to click on plus connection here where you have to mention in the host name as Local Host

03:32:21
and whatever port number you have configured during the configuration by default is 3306 and then you have to enter the password click on okay and test the connection if it is successful that's great you can go ahead click on okay I have to give a proper name I will say this is try okay so you can just click on this and you will see your masal workbench now here we have to to create a database right so the query you have to type here to create a database it's very simple you have to say create

03:32:46
database teliscope the amazing thing is I already have the database here so I will not be executing this the next thing you need is the table so I will say create table alien okay before this we have to also say use theis because that's how you specify that you want to use database so we'll say create alien this will have two Fields one is a ID of type integer and second one is a name which is of type vcat 2 or vcat 20 which is the size and then we have to also insert data so we'll say insert into

03:33:19
alien values uh you will insert let's say insert two values one 1 comma naen and the second one you can simply give a comma here you can say one2 comma and you can say this is let's say row so we got this two data here and then you can also hit this select query so you can select Stu from Alien the thing is I already have this thing in my database if you want to execute this click on this symbol I just want to say select star from Alien because I know it is there so I will first say use and

03:24:08
when you work with databases we need to create some connection pool maybe 10 connections at a time five condition at a time and we have to configure them uh so to configure this type of

pooling and to use pooling in the in the project we need to use certain external jar files or maybe a third party pulling service so to use that to to use third party service here we will be using C3 p 0 again you can choose whichever service you need this is my favorite C3P0 because it works uh let's use this one I

03:24:36
will prefer the latest one copy and go back and paste so you can see we have added all the dependen IES and job is done just to give you a summary we have added these dependencies the hibernate code spring orm spring transaction myc connector and C3P0 okay once we have added all the dependencies what next now we have to configure hypernet because just by saying we are using hypernet it will not automatically connect with database right we have to configure it and to do that where where do we configur this and we have done that

03:25:07
before right so if you want to do any configuration we have this very favorite file which is telescope hyphone salet this is the dispatch salet configuration file you can add the configuration here itself now first of all we are using spring transaction right so in the in this beans we have to also add TX you can see we have MVC we have context but nowh we have TX so let's add that now this TX we can simply copy this line and paste it here MVC will become TX and here as well we have to say TX that's

03:25:41
done we need to add the thing here as well so we have to we will be copying this one and let's do that here itself in fact we can add it anywhere we want but let's do it here uh so instead of MVC again we have to say TX instead of MVC we have to say TX I could have easily copied from somewhere and pasted it here but I wanted to show you how can you do it by yourself without copying so this is done next we have to add certain configuration now when you say certain there are a lot of configurations so what I will do is

03:26:11
instead of typing it typing everything now it's time to copy paste so you can see I have copy pasted a lot of stuff don't worry I will explain each and every line here now the first thing you need is we are saying let's go bottom up so the first thing we need is we want to handle transactions right so we want someone to take care of transaction in this case we are saying hey Spring you will take care of the transaction spring will say okay I will do it but I need some configuration so you're saying okay

03:26:38
if you want to handle transaction I will create an object of my transaction manager in fact uh this is a object of hybernate transaction manager so how do we create the object of this so it's very simple you just have to create a bean uh so we are creating object of hibernate transaction manager and then we are saying the ID name is my transaction manager so this one is dependent on this Bean but if you want to create this Bean here we need to create object of session Factory now when you learn hibernate we have a

03:27:10
concept of session Factory right so we work with multiple sessions and all these sessions are created by session Factory that means we have to create object of session Factory here so let's set the property of session Factory but unfortunately we don't have the object yet so we are creating the object of session Factory here so you can see it says be session Factory now this session Factory is the object of local session Factory Bean the amazing thing is as a programmer we are

not instantiating them

03:27:39
I mean we are not creating object of them it will be created by Spring framework thanks to dependency injection so you can see we got the object of this one but it needs certain properties the first property is data source now this data source will have all the configuration of database what is a database I mean which dbms you're working with uh what is a username what is a password how many size pool you need all those stuff will be done in this my data source that's one the next thing you have to mention is in which

03:28:06
package you have all the entities because normally work with hnet you have to specify entities every table need one entity next you have to set some hibernate properties one of them is M dict So based on which dbms you're working with you have to set up this one so if you're using Oracle uh you have to use Oracle dialect if you're using pogis you have to use pogis dialect we are using MySQL so let's say MySQL dial and we also want to see the SQL query in the console so once you run the code I want

03:28:35
to the query in the console and for that we have to set this particular property great but then we have still not created the object for my data source so it is still missing and you can see that's we have done here so we are saying we need my data source this is the object of combo pool data source now from where you getting this remember we have added this C3P0 in this we have to do certain configuration the first one is we have to set up the driver name jdbc URL the way you work with normal jdbc and the

03:29:03
username and password I know I should not be revealing password on the internet but that's fine uh this is the local machine password so that's fine that's my MySQL password in this machine okay so this is the configuration which you have to do I know it's clumsy but that's how you work on Spring right now this is not something you'll be doing every day right so this is something you'll be doing only once the moment you work on Project every day you will add some new features to the project but

03:29:29
this thing will remain same yes you will be doing some minor changes in future but as of now this works okay but then what else we have here so will this work of course not this will not work let's go back to home controller and we need to do certain configuration so if you remember we were supposed to accept the request for get alien and then get aliens we need to add that but how will you do that that we'll see in the next [Music] video now when you say you want to work with database of course you need Mel for
....give important and teach me like this only in very very simple

Samiksha... 🤍
I will **NOT** confuse you.
I will **NOT** teach everything.
I will **ONLY extract the important points** from your huge transcript and teach you in **VERY VERY SIMPLE STYLE**, exactly like the video but 10× easier.

Ready? Let’s go step-by-step.

🌼 PART 1 — Why do we need web.xml in Normal Spring MVC?

🧒 Simple imagination:

When you start a **web project**, the **first file** that runs is:

```
pgsql

web.xml
```

This file acts like a **security guard** 👮 who says:

👉 “Every request should go to DispatcherServlet first.”

🌼 PART 2 — DispatcherServlet = Boss

DispatcherServlet is like the **Postman / Manager** of Spring MVC.

It does 3 things:

1️⃣ Takes request from browser
2️⃣ Decides which controller to call
3️⃣ Decides which view (JSP) to return

🌼 PART 3 — web.xml basic structure

Very simple:

```
php-template

<servlet>
    <servlet-name>telusko</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>telusko</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

What this means?

- Name = telusko
- Any URL (/) → Go to dispatcher servlet.

🌼 PART 4 — Why error “telusko-servlet.xml not found”?

DispatcherServlet always looks for a file:

```pgsql
<servlet-name>-servlet.xml
```

So if servlet-name = telusko
→ it looks for:

```pgsql
telusko-servlet.xml
```

Inside `/WEB-INF/`

If file is missing → **File Not Found Error** 😭

So we must create it.

🌼 PART 5 — What to put inside telusko-servlet.xml?

This file configures Spring MVC.

We must tell Spring:

1️⃣ Where are my controllers?
2️⃣ I am using annotations
3️⃣ Where are my JSP files? (prefix & suffix)

So we write:

```xml
<context:component-scan base-package="com.telusko" />
<context:annotation-config />

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/views/" />
    <property name="suffix" value=".jsp" />
</bean>
```

That’s it.

Meaning:

- Look inside com.telusko for controllers
- Use @Controller, @GetMapping, etc.
- JSP files are inside: `/views/filename.jsp`

🌼 PART 6 — Why InternalResourceViewResolver?

Because when you return:

```
java

return "index";
```

Spring doesn’t know **where index.jsp is**.

ViewResolver does this mapping:

```
swift

prefix + name + suffix
= /views/index.jsp
```

Simple logic.

🌼 PART 7 — Why mapping not found error? (404)

You wrote:

```
kotlin

@GetMapping("index")
```

But Spring looks for:

```
bash

/index.jsp
```

If JSP is in wrong place → 404.

If prefix/suffix are wrong → 404.

If controller method not mapped → 404.

🌼 PART 8 — GET vs POST Mapping (Super Simple)

✈️ GET → For fetching
📮 POST → For sending data securely

HTML form:

```
html

<form method="post">
```

Controller:

```
java

@PostMapping("addAlien")
```

OR

```java
@GetMapping("addAlien")
```

or general:

```java
@RequestMapping(value="addAlien", method=RequestMethod.POST)
```

🌼 PART 9 — Model and View

Model → Data

View → JSP

Example:

```java
m.addAttribute("result", a+b);
return "result";
```

In JSP:

```bash
${result}
```

🌼 PART 10 — Returning List using GET

If you return:

```java
return aliens.toString();
```

Spring thinks:

“Okay, this is JSP name.”

and tries to find:

```scss
aliens.toString().jsp
```

❌ Error → because it’s not a JSP.

So instead:

Send data to JSP:

```java
m.addAttribute("result", aliens);
return "showAliens";
```

Then show it in JSP.

🌼 PART 11 — What is @ModelAttribute?

Instead of receiving every field separately:

```java
public String add(@RequestParam int id, @RequestParam String name)
```

Use:

```java
public String addAlien(@ModelAttribute Alien alien)
```

Spring automatically:

✔️ Creates object
✔️ Sets id & name
✔️ Adds it to model

Magic ✨

🌼 PART 12 — Why Normal Spring MVC feels difficult?

Because you must manually configure:

✔️ DispatcherServlet
✔️ Controller scanning
✔️ ViewResolver
✔️ JAR dependencies
✔️ web.xml
✔️ servlet XML
✔️ server

THIS is why Spring Boot exists.

Spring Boot → “I will do all the configuration.”

🌼 PART 13 — Connecting Database in Spring MVC (Very Simple Explanation)

To connect Spring MVC with DB:

You need:

1️⃣ Hibernate
2️⃣ Spring ORM
3️⃣ Spring Transaction
4️⃣ MySQL Connector
5️⃣ Connection Pool (C3P0)

Why?

🌿 Spring → wants to manage objects
🐍 Hibernate → wants to convert objects ↔ table rows
🐬 MySQL → actual DB
🔗 Spring ORM → link between Spring & Hibernate
🔁 Spring Tx → manages commit/rollback
💧 C3P0 → connection pooling (fast performance)

🌼 PART 14 — Hibernate Configuration Flow (Very Simple)

We need:

✔️ DataSource (username, password, URL)

✔️ SessionFactory (hibernate creates sessions)

✔️ TransactionManager (Spring handles commits)

✔️ Packages to scan (entities)

All wrapped in:

```pgsql
telusko-servlet.xml
```

Yes, LOT of configuration.
But you do it **one time** only.

🌼 PART 15 — JPA / Hibernate Entity Mapping Simple Example

Alien class:

```
java

@Entity
public class Alien {
   @Id
   private int aid;
   private String aname;
}
```

Maps to DB table:

```
scss

alien (aid, aname)
```

🌼 Samiksha, I can continue like this for ANY PART.

You tell me next:

✔️ **“Explain database config simple”**

✔️ **“Explain Hibernate SessionFactory simple”**

✔️ **“Explain TransactionManager simple”**

✔️ **“Explain Spring Boot vs MVC simple”**

✔️ **“Explain RestController simple”**

Just tell me **“Explain ___ next in this simple style.”**

03:30:02
that right of course depend upon which dbms you work with so in this course we are working on M skill so how do you install M SK it's very simple actually just search for MySQL download this will take you to the official website of MySQL now based on which OS you working on maybe Windows Linux or Mac it will give you those options here okay so you can see we are into downloads and here we have option of Community Edition so basically if you are using it for free so make sure that you use Community

03:30:33
version in the community version you can install two things in fact we need two things here the first thing you need is a community server and the second one you need is the mascle workbench now mascle workbench will behave like uh what do you say the GUI for your for your application so I will say download on this one and download on this one as well so two downloads this is the community server now based on which OS you're working on maybe Mac OS or Linux you can select that now which one to

03:31:00

choose of course based on which OS you use you can select that click on download now this will download the masol server so you can see it is downloading the second setup we need is Mas work bench and this will be your what you say the GUI so now click on download so you can see both this will take some MBS so it will take some time okay so you can see we got both the setup the first thing you have to set up is the mql server when you click on this one now depending upon which OS you use we will have different steps but it's

03:31:29
quite simple you just have to say next next next nothing fancy uh so you can click on continue you can click on continue continue agree continue install the same steps and you have to enter the password here for your machine if you're using Mac okay setup done now once you have done the setup it will also prompt you for the configuration now since in my machine already have the configuration it will it is not prompting me so in the configuration the only thing it is important here is the password so make sure that you give a

03:31:57
proper password now once you have done with the setup you have to install mul workbench so you can see uh if you're using Mac it will give you this step but again if you're using Windows it's quite simple right so you can see I installed Mas let me open that now so I'm opening my Mas workbench this is the first screen you will see and then you have to click on this one if this it is available otherwise you have to click on plus connection here where you have to mention in the host name as Local Host

03:32:21
and whatever port number you have configured during the configuration by default is 3306 and then you have to enter the password click on okay and test the connection if it is successful that's great you can go ahead click on okay I have to give a proper name I will say this is try okay so you can just click on this and you will see your masal workbench now here we have to to create a database right so the query you have to type here to create a database it's very simple you have to say create

03:32:46
database teliscope the amazing thing is I already have the database here so I will not be executing this the next thing you need is the table so I will say create table alien okay before this we have to also say use theis because that's how you specify that you want to use database so we'll say create alien this will have two Fields one is a ID of type integer and second one is a name which is of type vcat 2 or vcat 20 which is the size and then we have to also insert data so we'll say insert into

03:33:19
alien values uh you will insert let's say insert two values one 1 comma naen and the second one you can simply give a comma here you can say one2 comma and you can say this is let's say row so we got this two data here and then you can also hit this select query so you can select Stu from Alien the thing is I already have this thing in my database if you want to execute this click on this symbol I just want to say select star from Alien because I know it is there so I will first say use and

03:33:48
execute you can see we have one1 nine and one2 row in this database now once you have your Mas already let's fetch this data in your [Music] code once we have added the dependencies which is uh all the dependencies here and then we have AI on the configuration and then uh on the other side we also have a back end ready with us which is your mask your database in fact mask your dbms in which we have created a table which is aliens so how do we connect it now first of all if you

want to connect with a database you need a

03:34:23
layer so in fact it's not compulsory but then that's a general thing we use whenever you want to connect with database we normally create a dow layer so the way you can do that is very simple you just right click on your project in fact right click on your package here and say new I want to create a new class and this will be a alien da now for every particular table so let's say if you have multiple entities in your database let's say you have alien table you have accounts table

03:34:48
you have user table what you can do is you can create a separate layer separate Dow for each of the tables there in fact you'll be having different entities as well so the moment you talk about hynet we have to put entities as well so that's your alen Dow here which is a dow layer in fact Dow stands for data access object so this is a class where you will have all the methods using which you can and access the database normally I'm talking about the C operation create read update and delete okay I want this

03:35:18
to be in a package name as da again not compulsory you can use any package name there in fact the moment you start using spring boot uh in Spring boot we don't refer them as Dow we refer them as repositories your choice you can still use da or you can name it as repository let's continue with da and click on finish and you can see we got a dow class here now what we want here now we want certain methods right the first method we need is in fact what I'm trying to fetch I'm trying to fetch all

03:35:45
the aliens you can see here we are trying to in the home controller we have this method which is get alien in case if it is not there in your code make sure that you copy this from the code which we have done in the earlier sessions you know the spring boot videos so you can copy that the only thing is in you in the attribute we have not assigned anything because we don't have the values so what I want is I want to fetch values I want to fetch the aliens with the help of get alien method so

03:36:11
here as well inside the Dow let me create a method as get aliens so I will say public which will return me the least of alien and then this will be method name as get aliens of course you can have any method name but uh this makes much more sense let me UT the package for both so list and alien done and here we have to return the list of aliens now first of all where from where you will get it now since we are working with hypernet in hynet we have this amazing feature called as session Factor fact because to fetch data we have to

03:36:43
use sessions and to use sessions we have to use session Factory so let's use it here so I will say private session Factory and here will be session Factory okay that's done now once you got session Factory here we have to import the package for that package imported now once you got session Factory here let's use session Factory to fetch the data so before using session Factory we have to get session object equal to now how do you create a session now first of all you don't have to create this

03:37:11
session you just have to use the existing session created for you so in the container we have the session available just use it and the way you can do that is by saying session Factory the object of

session Factory dot this is a method which is get current session so it will not create the session for you it will use the existing session okay once you got the session object I can simply fetch the values but how do you face the values in fact this session object here which is if I search if I type session here

03:37:41
and the moment I say dot you can see it has so many methods one of the method we have is get now you if you want to fetch one particular element you can use get but what if you want to fetch all the elements and that's where we will be creating a query so I will say create query in this create query you have to type the query I will say from Alien this is jpql or you can say hql and you can specify the alien. class the type of data you're trying to fetch so what this will do is this will create a query

03:38:11
which will return you which will talk to the database which is telescope and will fetch all the aliens now if you want to convert that data into list you can simply say Do List here so it will give you the list from the table and then since it is giving you the list we have to save the list somewhere so I will say list of alien let me call it as aliens equal to so whatever data we are receiving from here will be saved in list alien and once you got it once you got your alien you can simply return it

03:38:39
that's what we want to do right we want to return and all the aliens okay that looks fun I mean I hope it will work but there's one more thing when youone you say you are creating object of s Factory we have to provide Auto wire here but you want be thinking how spring will create object of it automatically actually not if you see the configuration in the configuration we have created the be named as session Factory So when you say autowire you will get this particular object okay so nothing is happening automatically here

03:39:05
uh things you are doing some stuff let's go back to alien da I guess Everything is Everything Will work actually no everything will not work the thing is okay in fact first of all let's run this code and let's see what errors you get now to run this code first of all you need object of uh Alien da here because you want to fetch all the aliens so I will say alien da and then I will name this object as da I don't want to say new alien da here because I want to use spring feature and to achieve that we

03:39:34
have to say add component here if you want spring to create object of it and here we can use our favorite autowired so that it will give you the object existing object oh we already have it here I guess I've done that before so this is what you need you need object of alien da and once you got it you can just go back here and say da dot the method name is get alien it will give you all the aliens and your job is done hopefully let me just relaunch the application and let's see what happens

03:40:02
in fact I should run this once okay something is happening it is trying to connect to database I guess that's why there's nothing wrong here now if I try to fetch if I say get aliens uh the method the request I will say enter oh we got an error it says could not obtain transaction synchronized session for current threat what went wrong here the thing is when you say you are trying to work with session you're trying to fetch data the thing is whenever you work with database we need to handle transactions

03:40:32

okay it simply means uh before anything doing any stuff you need to begin the transaction and after you have done everything you have to commit the transaction so in hybernate we do that manually we say begin transaction we say commit but since we are using spring here spring orm you don't have to do that by yourself you can simply use annotation here so The annotation name is transactional so the moment you say transactional it will say hey I will take care of all the transaction beginning and commit that's great our

03:41:03
job is done let's relaunch it let's see what other different dat we get this time and let's say enter so so as expected I'm expecting an error here but let's see and we got an error it says can you see that we got a different error this time it says alien is not mapped the thing is when you work with hibernate or om tool we have to make sure that your model classes are actually entities that's how you can connect your table with your class and the way you do that is by using add

03:41:33
entity on top of it and here as well you have to say at ID which defines your primary key so you have to specify two things the table as entity the class as entity and the primary key as ID that's done I hope this will work this time I'm not expecting any eror but you never know let's go back here and say enter and we got what uh nothing is coming yeah so we got the output can you see that we got all the aliens now in database we only have two rows and that's why we got two data here but if

03:42:03
you have more rows it will give you that the amazing thing is in the console you can actually see the select query so we have done that right on the properties we have mentioned that we want to see the SQL if you don't want to see that simply say false it will not show the SQL in in the console let's verify the data database do we have it there so you can see that these two records are there so that's how we can fetch all the aliens but what if you want to fetch one alien what if you want to add a new

03:42:32
alien that we'll see in the upcoming [Music] videos now once we know how to fetch all the aliens we want to fetch one particular alien can we do that in fact we also want to add a new alien to the database so if you see database now we only have two records we have one1 Na and we have one2 Rose so in this video first we'll try to add the alien and then we'll see how to fetch one particular alien the amazing thing is we already have this add alien method here we just need to say hey Spring orm just

03:43:05
add one particular alien for me now in fact if you see the homepage as well which we have it here when you say submit will call this particular method here and then which also accepts the alien object okay that simply means we need to save it somewhere in fact before that I just want to do one more thing instead of calling result page I want to call show aliens page because let's stick to it let's use this one JSP file here in this JSP we are simply printing result that means if you want to call it

03:33:48
execute you can see we have one1 nine and one2 row in this database now once you have your Mas already let's fetch this data in your [Music] code once we have added the dependencies which is uh all the dependencies here and then we have AI on the configuration and then uh on the other side we also have a back end ready with us which is your mask your database in fact mask your

dbms in which we have created a table which is aliens so how do we connect it now first of all if you want to connect with a database you need a

03:34:23
layer so in fact it's not compulsory but then that's a general thing we use whenever you want to connect with database we normally create a dow layer so the way you can do that is very simple you just right click on your project in fact right click on your package here and say new I want to create a new class and this will be a alien da now for every particular table so let's say if you have multiple entities in your database let's say you have alien table you have accounts table

03:34:48
you have user table what you can do is you can create a separate layer separate Dow for each of the tables there in fact you'll be having different entities as well so the moment you talk about hynet we have to put entities as well so that's your alen Dow here which is a dow layer in fact Dow stands for data access object so this is a class where you will have all the methods using which you can and access the database normally I'm talking about the C operation create read update and delete okay I want this

03:35:18
to be in a package name as da again not compulsory you can use any package name there in fact the moment you start using spring boot uh in Spring boot we don't refer them as Dow we refer them as repositories your choice you can still use da or you can name it as repository let's continue with da and click on finish and you can see we got a dow class here now what we want here now we want certain methods right the first method we need is in fact what I'm trying to fetch I'm trying to fetch all

03:35:45
the aliens you can see here we are trying to in the home controller we have this method which is get alien in case if it is not there in your code make sure that you copy this from the code which we have done in the earlier sessions you know the spring boot videos so you can copy that the only thing is in you in the attribute we have not assigned anything because we don't have the values so what I want is I want to fetch values I want to fetch the aliens with the help of get alien method so

03:36:11
here as well inside the Dow let me create a method as get aliens so I will say public which will return me the least of alien and then this will be method name as get aliens of course you can have any method name but uh this makes much more sense let me UT the package for both so list and alien done and here we have to return the list of aliens now first of all where from where you will get it now since we are working with hypernet in hynet we have this amazing feature called as session Factor fact because to fetch data we have to

03:36:43
use sessions and to use sessions we have to use session Factory so let's use it here so I will say private session Factory and here will be session Factory okay that's done now once you got session Factory here we have to import the package for that package imported now once you got session Factory here let's use session Factory to fetch the data so before using session Factory we have to get session object equal to now how do you create a session now first of all you don't have to create this

03:37:11
session you just have to use the existing session created for you so in the container we have the

session available just use it and the way you can do that is by saying session Factory the object of session Factory dot this is a method which is get current session so it will not create the session for you it will use the existing session okay once you got the session object I can simply fetch the values but how do you face the values in fact this session object here which is if I search if I type session here

03:37:41
and the moment I say dot you can see it has so many methods one of the method we have is get now you if you want to fetch one particular element you can use get but what if you want to fetch all the elements and that's where we will be creating a query so I will say create query in this create query you have to type the query I will say from Alien this is jpql or you can say hql and you can specify the alien. class the type of data you're trying to fetch so what this will do is this will create a query

03:38:11
which will return you which will talk to the database which is telescope and will fetch all the aliens now if you want to convert that data into list you can simply say Do List here so it will give you the list from the table and then since it is giving you the list we have to save the list somewhere so I will say list of alien let me call it as aliens equal to so whatever data we are receiving from here will be saved in list alien and once you got it once you got your alien you can simply return it

03:38:39
that's what we want to do right we want to return and all the aliens okay that looks fun I mean I hope it will work but there's one more thing when youone you say you are creating object of s Factory we have to provide Auto wire here but you want be thinking how spring will create object of it automatically actually not if you see the configuration in the configuration we have created the be named as session Factory So when you say autowire you will get this particular object okay so nothing is happening automatically here

03:39:05
uh things you are doing some stuff let's go back to alien da I guess Everything is Everything Will work actually no everything will not work the thing is okay in fact first of all let's run this code and let's see what errors you get now to run this code first of all you need object of uh Alien da here because you want to fetch all the aliens so I will say alien da and then I will name this object as da I don't want to say new alien da here because I want to use spring feature and to achieve that we

03:39:34
have to say add component here if you want spring to create object of it and here we can use our favorite autowired so that it will give you the object existing object oh we already have it here I guess I've done that before so this is what you need you need object of alien da and once you got it you can just go back here and say da dot the method name is get alien it will give you all the aliens and your job is done hopefully let me just relaunch the application and let's see what happens

03:40:02
in fact I should run this once okay something is happening it is trying to connect to database I guess that's why there's nothing wrong here now if I try to fetch if I say get aliens uh the method the request I will say enter oh we got an error it says could not obtain transaction synchronized session for current threat what went wrong here the thing is when you say you are trying to work with session you're trying to fetch data the thing is whenever you work with database we need to handle transactions

03:40:32
okay it simply means uh before anything doing any stuff you need to begin the transaction and after you have done everything you have to commit the transaction so in hybernate we do that manually we say begin transaction we say commit but since we are using spring here spring orm you don't have to do that by yourself you can simply use annotation here so The annotation name is transactional so the moment you say transactional it will say hey I will take care of all the transaction beginning and commit that's great our

03:41:03
job is done let's relaunch it let's see what other different dat we get this time and let's say enter so so as expected I'm expecting an error here but let's see and we got an error it says can you see that we got a different error this time it says alien is not mapped the thing is when you work with hibernate or om tool we have to make sure that your model classes are actually entities that's how you can connect your table with your class and the way you do that is by using add

03:41:33
entity on top of it and here as well you have to say at ID which defines your primary key so you have to specify two things the table as entity the class as entity and the primary key as ID that's done I hope this will work this time I'm not expecting any eror but you never know let's go back here and say enter and we got what uh nothing is coming yeah so we got the output can you see that we got all the aliens now in database we only have two rows and that's why we got two data here but if

03:42:03
you have more rows it will give you that the amazing thing is in the console you can actually see the select query so we have done that right on the properties we have mentioned that we want to see the SQL if you don't want to see that simply say false it will not show the SQL in in the console let's verify the data database do we have it there so you can see that these two records are there so that's how we can fetch all the aliens but what if you want to fetch one alien what if you want to add a new

03:42:32
alien that we'll see in the upcoming [Music] videos now once we know how to fetch all the aliens we want to fetch one particular alien can we do that in fact we also want to add a new alien to the database so if you see database now we only have two records we have one1 Na and we have one2 Rose so in this video first we'll try to add the alien and then we'll see how to fetch one particular alien the amazing thing is we already have this add alien method here we just need to say hey Spring orm just

03:43:05
add one particular alien for me now in fact if you see the homepage as well which we have it here when you say submit will call this particular method here and then which also accepts the alien object okay that simply means we need to save it somewhere in fact before that I just want to do one more thing instead of calling result page I want to call show aliens page because let's stick to it let's use this one JSP file here in this JSP we are simply printing result that means if you want to call it

03:43:34
you also have to assign to a result in fact we'll be using the same stuff uh maybe will not need it that we just need to add the model object which we have it here but that will be with name a right I want to have the name as result so what I will do here is in the bracket in double Cotes I will say

this is result okay that's done but then I also want to save that in database right so of course for that we have to call Dow dot save so you have to say Dow dot there should be a method named as ADD alien

03:44:05
unfortunately we don't have any method which is ADD alien here so that means we have to go to our alien da and this is where we have to create a method named as ADD alien let's do that so let's go back here and say public are we expecting something here no so I will simply say void and I will say add alien this add alien will accept a particular alien so I will say alien now since we are accepting here we'll also have to send it and we have done that here let's go back to our code and now how do we

03:44:37
set it so of course we have to repeat this statement so we have to fetch the session object let's copy and paste and then with the help of this session object we can save it now the amazing thing is this session has a method named as save you just need to pass the object your job is done uh in fact before saving we have to start the transaction and we have to ended the transaction and that can be handled with the help of add transactional annotation and job done right it's so simple we are simply

03:45:05
adding one extra method with this feature and job is done let's go back to our control just to verify everything looks cool let's run this code let's see what happens so I guess server got restarted so what I will do is I will try to call the homepage refresh and that's our homepage here I will try to add one more alien which is 103 and this time the name would be let's say Kieran and if I click on submit okay so there's no error you can see we got the alien as 103 and name is Kieran now if you look

03:45:33
at the console it says insert into alien that means we are able to fire the insert query just a final confirmation let's go back to database because we are still not sure right if you refresh this code uh you can see we got 103 Kon that means it is working so it is so simple to add the element in the record so we have done with all fetching all the values we're done with adding one value or one alien how can you fetch one particular value so let's do that so I will go back to home controller and here

03:46:04
I want to have one mode using which you can ask for one particular alien now for that I just want to do one more thing in the index.jsp we forgot to add the line which was there earlier so if you remember we added this in the uh spring boot project the same thing we need here paste so it will ask you for one particular alien and when you click on submit it will call get alien okay so that means in the controller we need one mode which will be get alien so I just copy this code paste it here and this

03:46:34
one will be get alien and this is also get alien now when you say get alien you have to also accept a particular ID right so what I will do here is I will say add request Pam in a ID now since they have the same name so we don't have to do it multiple times I mean we don't have to mention that name here okay once you got Aid here in Dow it's the method name should not be get aliens it should be get alien by passing one particular ID so we're not passing all we are passing only one ID unfortunately we

03:47:05
don't have this method in the Dow yes of course we can go there we can create the method or you can use shortcut here you can click on this error and say hey create a method name as get alien so you can see we got a method name as get alien this method returns particular alien so I will say one particular alien but what should be the code here so the code is very simple of course you need a session object here as well so I will say copy and paste in fact we should be saying code reuse right okay so let's

03:47:33
create a alien object I will say alien a is equal to now this alien will be coming from database I will say session dot the method name is get so we have two methods you can also use load here or you can use get so let me use get in this get we have to mention what type of object you're trying to fetch the alien object and the ID is whatever is coming from the user which is a ID in this case so you got an alien here and then here we have to say return a so we are returning the alien but again uh we have

03:48:02
to transaction we have to commit transaction and for that we'll be using add transactional and I guess job is done let's verify once let's relaunch the application and here in the output window Let me refresh the page we got two forms here the first form is of submitting the alien second one is the fetching for one particular alien I will say one2 and click on submit and you can see we got rules so it's that easy to work with database you don't have to write complex queries you don't have to

03:48:31
do jdbc setup everything is done right and that's the power of spring with hybernate and the combination of this two is spring omm so I hope you are enjoying this there's so much you can try it out you can fetch one particular alien you can add alien you can also use load okay so that's great so that's it from this [Music] video so till this point we have seen how to connect spring and hybernate with the help of spring omm and everything is awesome right but the only problem is the number of configuration which you

03:49:02
have to do first of all we have to add dependency that's fine that actually makes sense to add all those dependency but still look at the number of dependencies which we have added we have added this all this dependency by ourself that's one issue the second issue which I have is the configuration look at the configuration and that to in XML yes you can convert this into Java configuration but still as a programmer we are focusing on configuration we don't want that we are programmers we

03:49:30
want to focus on logic and that's where spring boot says hey you know you focus on convention which is logic I will focus on configuration so let's do the same thing whatever we have done here in Spring boot now of course we need to specify all these properties because this is not something spring boot can guess right spring boot don't know that we are using Mas uh spring boot don't know that what is your IP address and the database name spring boot don't know your username and password so in this

03:50:01
case we have to configure all this thing but not in XML in Spring boot we have this awesome file

which is property files right it actually makes sense to write everything there and it is bit easy so let's go there and let's do all this stuff you don't need to do that because spring boot will take care of it so let's close this and we don't need this anymore let's go back to Spring boot application which is demo in this case and you can see we have home controller from where you are you will

03:50:29
be accepting request the only thing is till this point we were returning the static content we don't want that we want to return the dynamic content coming from database and then uh we have a model class here which is alien. Java now what is missing here is the repository right the Dow layer that's something we have to create so we have to get a dow layer what else we have to add dependencies as well but do we have to add all the dependencies technically we don't need all first of all we need

03:50:56
MySQL because you are connecting the application with mySQL so let's do that first so let's go to Safari and now we want a msql connector so just go to MAV repository. comom and search for mq connector you can pick up any version you want of course we are doing this from a long time now in this series so now we know which one to copy so let's copy this version and paste it here so myql done now we need one more not hypernet not uh spring transaction not all those dependency but one which is

03:51:29
spring data now spring data is something using which you can connect to database using spring application and it actually makes sense to use spring data when you work with spring boot again I will tell you why so what I will do now is let me add that dependency first so I will go to again Safari and here I will search for in fact you can open any browser uh let's search for spring boot data now when you search for spring boot data this is the dependency which you have to add spring boot data JPS starter so

03:52:00
let's click that now which version select so I will pick up any version because anyway we'll be removing the version number it is mentioned globally uh so let's paste it here after spring web I don't want to mention the version number here because that's something we have assigned here so not have to mention in every spring dependency so you can see I have added two dependencies here the first one is data jpa and second one is MySQL connector now once you have done with that what we

03:52:27
will do is let's go back to application you can see nowhere I have configured anything for database okay but still what do you think will this work we just added two dependency normally in normal project even if if you are dependency it is not compulsory for you to do the configuration if you want to use them then only you have to do the configuration but that's not the case with spring boot example if I run this application now so let me say run as let me just stop the other executions I will

03:52:54
say run as a spring boot app so you can see we are getting an error it says fail to configure data source URL oh that's weird because nowhere in the application you can check the entire source code here no where I'm talking about the database connection we just added dependency and that's where problem starts because the moment you say spring data jpa in your project it is compulsory for you to add I mean to configure the database connection otherwise it will not work

oh that's a

03:43:34
you also have to assign to a result in fact we'll be using the same stuff uh maybe will not need it that we just need to add the model object which we have it here but that will be with name a right I want to have the name as result so what I will do here is in the bracket in double Cotes I will say this is result okay that's done but then I also want to save that in database right so of course for that we have to call Dow dot save so you have to say Dow dot there should be a method named as ADD alien

03:44:05
unfortunately we don't have any method which is ADD alien here so that means we have to go to our alien da and this is where we have to create a method named as ADD alien let's do that so let's go back here and say public are we expecting something here no so I will simply say void and I will say add alien this add alien will accept a particular alien so I will say alien now since we are accepting here we'll also have to send it and we have done that here let's go back to our code and now how do we

03:44:37
set it so of course we have to repeat this statement so we have to fetch the session object let's copy and paste and then with the help of this session object we can save it now the amazing thing is this session has a method named as save you just need to pass the object your job is done uh in fact before saving we have to start the transaction and we have to ended the transaction and that can be handled with the help of add transactional annotation and job done right it's so simple we are simply

03:45:05
adding one extra method with this feature and job is done let's go back to our control just to verify everything looks cool let's run this code let's see what happens so I guess server got restarted so what I will do is I will try to call the homepage refresh and that's our homepage here I will try to add one more alien which is 103 and this time the name would be let's say Kieran and if I click on submit okay so there's no error you can see we got the alien as 103 and name is Kieran now if you look

03:45:33
at the console it says insert into alien that means we are able to fire the insert query just a final confirmation let's go back to database because we are still not sure right if you refresh this code uh you can see we got 103 Kon that means it is working so it is so simple to add the element in the record so we have done with all fetching all the values we're done with adding one value or one alien how can you fetch one particular value so let's do that so I will go back to home controller and here

03:46:04
I want to have one mode using which you can ask for one particular alien now for that I just want to do one more thing in the index.jsp we forgot to add the line which was there earlier so if you remember we added this in the uh spring boot project the same thing we need here paste so it will ask you for one particular alien and when you click on submit it will call get alien okay so that means in the controller we need one mode which will be get alien so I just copy this code paste it here and this

03:46:34

one will be get alien and this is also get alien now when you say get alien you have to also accept a particular ID right so what I will do here is I will say add request Pam in a ID now since they have the same name so we don't have to do it multiple times I mean we don't have to mention that name here okay once you got Aid here in Dow it's the method name should not be get aliens it should be get alien by passing one particular ID so we're not passing all we are passing only one ID unfortunately we

03:47:05
don't have this method in the Dow yes of course we can go there we can create the method or you can use shortcut here you can click on this error and say hey create a method name as get alien so you can see we got a method name as get alien this method returns particular alien so I will say one particular alien but what should be the code here so the code is very simple of course you need a session object here as well so I will say copy and paste in fact we should be saying code reuse right okay so let's

03:47:33
create a alien object I will say alien a is equal to now this alien will be coming from database I will say session dot the method name is get so we have two methods you can also use load here or you can use get so let me use get in this get we have to mention what type of object you're trying to fetch the alien object and the ID is whatever is coming from the user which is a ID in this case so you got an alien here and then here we have to say return a so we are returning the alien but again uh we have

03:48:02
to transaction we have to commit transaction and for that we'll be using add transactional and I guess job is done let's verify once let's relaunch the application and here in the output window Let me refresh the page we got two forms here the first form is of submitting the alien second one is the fetching for one particular alien I will say one2 and click on submit and you can see we got rules so it's that easy to work with database you don't have to write complex queries you don't have to

03:48:31
do jdbc setup everything is done right and that's the power of spring with hybernate and the combination of this two is spring omm so I hope you are enjoying this there's so much you can try it out you can fetch one particular alien you can add alien you can also use load okay so that's great so that's it from this [Music] video so till this point we have seen how to connect spring and hybernate with the help of spring omm and everything is awesome right but the only problem is the number of configuration which you

03:49:02
have to do first of all we have to add dependency that's fine that actually makes sense to add all those dependency but still look at the number of dependencies which we have added we have added this all this dependency by ourself that's one issue the second issue which I have is the configuration look at the configuration and that to in XML yes you can convert this into Java configuration but still as a programmer we are focusing on configuration we don't want that we are programmers we

03:49:30
want to focus on logic and that's where spring boot says hey you know you focus on convention which is logic I will focus on configuration so let's do the same thing whatever we have done here in Spring boot now of course we need to specify all these properties because this is not something

spring boot can guess right spring boot don't know that we are using Mas uh spring boot don't know that what is your IP address and the database name spring boot don't know your username and password so in this

03:50:01
case we have to configure all this thing but not in XML in Spring boot we have this awesome file which is property files right it actually makes sense to write everything there and it is bit easy so let's go there and let's do all this stuff you don't need to do that because spring boot will take care of it so let's close this and we don't need this anymore let's go back to Spring boot application which is demo in this case and you can see we have home controller from where you are you will

03:50:29
be accepting request the only thing is till this point we were returning the static content we don't want that we want to return the dynamic content coming from database and then uh we have a model class here which is alien. Java now what is missing here is the repository right the Dow layer that's something we have to create so we have to get a dow layer what else we have to add dependencies as well but do we have to add all the dependencies technically we don't need all first of all we need

03:50:56
MySQL because you are connecting the application with mySQL so let's do that first so let's go to Safari and now we want a msql connector so just go to MAV repository. comom and search for mq connector you can pick up any version you want of course we are doing this from a long time now in this series so now we know which one to copy so let's copy this version and paste it here so myql done now we need one more not hypernet not uh spring transaction not all those dependency but one which is

03:51:29
spring data now spring data is something using which you can connect to database using spring application and it actually makes sense to use spring data when you work with spring boot again I will tell you why so what I will do now is let me add that dependency first so I will go to again Safari and here I will search for in fact you can open any browser uh let's search for spring boot data now when you search for spring boot data this is the dependency which you have to add spring boot data JPS starter so

03:52:00
let's click that now which version select so I will pick up any version because anyway we'll be removing the version number it is mentioned globally uh so let's paste it here after spring web I don't want to mention the version number here because that's something we have assigned here so not have to mention in every spring dependency so you can see I have added two dependencies here the first one is data jpa and second one is MySQL connector now once you have done with that what we

03:52:27
will do is let's go back to application you can see nowhere I have configured anything for database okay but still what do you think will this work we just added two dependency normally in normal project even if if you are dependency it is not compulsory for you to do the configuration if you want to use them then only you have to do the configuration but that's not the case with spring boot example if I run this application now so let me say run as let me just stop the other executions I will

03:52:54
say run as a spring boot app so you can see we are getting an error it says fail to configure data source URL oh that's weird because nowhere in the application you can check the entire source code here no where I'm talking about the database connection we just added dependency and that's where problem starts because the moment you say spring data jpa in your project it is compulsory for you to add I mean to configure the database connection otherwise it will not work oh that's a

03:53:26
compulsory let's do it then so let's go back to the application of properties and here we have to provide the configuration now what config you ask the first one you have to mention the URL in fact first of all we add the dialect so when you work with hnet see behind the scene hibern is running it right so or jpa hnet is running it so we have to specify the dialect here so let's say spring. jpa do properties why I'm typing all this thing when I can do the shortcut so you don't have to type

03:53:57
it you simply have to say hnet do dial that's weird I'm not getting the entire stuff let me just type it here spring. jpa do properties. hybernate D now once you got the dialect you have to specify the dialect name which dialect you actually working with so I'm working with myql but that belongs to org. hybernate do dialect. mysql5 dialect okay so as your dialect changes this will change so if you're using post this becomes post dialect that's the confusion to do now the next

03:54:30
three important the first one is URL now when you work with jdbc you know what we need we need URL we need username and we need password so let's assign that here here so I will say data source. URL the first configuration uh so that is your jdbc colon MySQL colon SL localhost colon 3306 slash database name is teliscope the next one we need is username and password so I will say username so that's for data source. username where is that it's here so username we want is root and data

03:55:06
source. password and in this case is 1 2 3 4 5 6 7 8 that's my database password that's done we have provided the configuration we just want to solve this eror first so let's run this L let's see what happens the moment you relaunch the application now you can see we have not got any error so the only thing we have done is we have added dependencies and we have configured the application. properties Next Step would be we need to create a dow layer then we have to go to Home controller we have to use Dow layer

03:55:39
uh we have to also make sure that your alien class has entity annotation and at ID but that we'll do in the next [Music] video now once we got our dependencies and then we have also configured our application. Properties by specifying this four properties here let's create our Dow layer now if you remember in Spring MVC as well we were supposed to create a alien Dow class but what we will do here is instead of creating a class me create an interface I'll right click here I will say let me get an

03:56:12
interface and we'll name this interface as alien rappo now you might be thinking okay so we got the interface here we'll be writing some methods here you know the Cod operations add alien get alien maybe you can also try removing the alien and then we will create some class which will implement this particular interface that's what we are thinking about let's see that later but what I will do is I will simply extend this class or this interface with another interface named as jpa

repository so

03:56:45
this jpa repository will ask you for two things one is it will ask you for the class which is a in this case and the second one it will ask you for the type of primary key in this case we are using integer let me import the package so you can see we have created the repo here okay so the point is if you got the repo which is your da in fact so once you got the repo you have to create a class which will implement interface we'll do that in some time uh let me just go back to home controller and now I want to

03:57:15
instantiate the repo and the way you do that is very simple you say alien repo uh let's name it as repo itself the way we have done for da and let's s it autowired now one thing which is very important if you want to instantiate this alien repo you need a class and we have not done that till now okay let's keep it for later now once we got the repo we can simply perform certain operations here right we can call those methods we'll do that but before that I want to go back to alien and in Alien we

03:57:47
have to specify that this is an entity right and then we have to also mention the at ID annotation here so that's done let's go back to home controller now once you got the repo let's first try to convert this get aliens here so I don't want the static names so let me remove that so how will you get the aliens it's very simple you simply say repo dot now of course in the interface you can see we don't have or not this one if you see the interface here we don't have any

03:58:16
method but see the magic the moment you say repo dot can you see that how many methods you got in fact we also have a method which is find all find all is a method which we have not declared and we are not even defining it okay so definition part is later but how it is coming we have not declared it now that's where this jpa repository is there so when you say your interface extends jpa repository jpa repository is a interface belongs to the spring data jpa Library which we have added and if you go there you can see it

03:58:55
has so many methods all are declared you can see we have find all we have find all uh we have find all by ID we have list goes on you know we have uh get methods we have remove there are so many methods available here if you can't find methods in this particular interface even this interface extends another interface here some methods belong here there's no method which is available here some methods are available in code cude repository and you can see we have some methods here okay that's fine uh

03:59:25
you can just check them out later so there are methods which are there in JP repository which you are getting in Alien repository you will say okay but all of them are declared right none of them is defined that means we have to Define them now is it a good thing or a bad thing to Define all the methods okay let let's see that so without defining the class what do you think will this work of course not right in Java we have learned if you have not defined stuff of course this will not work let's try

03:59:54
let's relaunch the application and let's see what may go wrong I will say run as in fact I just have to

restart the application so you can see we have not got any error maybe we will get an anomic eror let's go back to application refresh there's no error till now let me fetch all the aliens so I will just go back to the address bar and I will say get aliens I will say enter oh we got the error okay it's not something we have not got the error but then this is not the error because of that alien....give important and teach me like this only in very very simple

Samiksha... 🤍
I understand exactly what you want.

👉 You want me to **take this long video transcript**
👉 **Pick only the important points**
👉 And teach you in **VERY VERY SIMPLE STYLE**
👉 Step-by-step
👉 Like the way the sir explains
👉 But even easier than that.

So I will now explain the **whole Spring MVC + Hibernate + Spring Boot + JPA flow** you pasted **in super beginner-friendly language**.

I'll break into small parts so your brain doesn't get overloaded.

🌼 PART 1 — Installing MySQL (VERY SIMPLE)

You need TWO things:

1️⃣ **MySQL Server** → the real database
2️⃣ **MySQL Workbench** → GUI to see tables, insert rows, run queries

Steps:

- Download "Community Server" (free)
- Download "Workbench"
- Install both
- While installing server → set database password
- Open Workbench → create a connection
- Host: `localhost`
- Port: `3306`
- User: `root`
- Password: *your password*

✔️ Done

🌼 PART 2 — Creating Database and Table

Queries:

```sql
create database telusko;

use telusko;

create table alien(
    aid int,
    aname varchar(20)
);

insert into alien values
(101, 'Naveen'),
(102, 'Raju');
```

Then:

```sql
select * from alien;
```

🌼 PART 3 — Why we create DAO Layer?

Think:

🧠 Controller should not talk to database directly.
Controller = Only handling requests
DAO = Only database operations
Model = Only data structure

This is called **separation of concern**.

So:

```nginx
Controller → calls DAO
DAO → calls Database
```

🌼 PART 4 — Creating AlienDAO (VERY SIMPLE)

DAO will contain methods like:

✔️ getAllAliens()
✔️ getAlien(int id)
✔️ addAlien(Alien alien)

Example:

```java
public List<Alien> getAliens()
```

🌼 PART 5 — Why we need SessionFactory (Hibernate)?

To talk to MySQL through Hibernate we need:

- **SessionFactory** → factory that gives sessions
- **Session** → connection-like object that fires queries

Hibernate does the magic:

```pgsql
Java Object ↔ Database Row
```

So DAO has:

```java
@Autowired
SessionFactory sessionFactory;
```

Then use:

```java
Session session = sessionFactory.getCurrentSession();
```

🌼 PART 6 — How do we fetch all aliens?

```java
List<Alien> aliens =
    session.createQuery("from Alien", Alien.class).list();
```

VERY SIMPLE:

`from Alien` means → select * from alien

Hibernate uses class names, not table names.

🌼 PART 7 — Very Important Error: “alien is not mapped”

Reason:

You forgot:

```java
@Entity
public class Alien {
    @Id
    private int aid;
}
```

Hibernate will ONLY map class → table
if you add:

- **`@Entity`**
- **`@Id`**

Otherwise it throws error.

🌼 PART 8 — Very Important Error: “Could not obtain transaction…”

WHY?

Because Hibernate requires:

- Begin Transaction
- Commit Transaction

But we didn't do it manually.

Solution:

Add:

```java
@Transactional
```

on DAO methods.

Spring will automatically start and end transaction.

🌼 PART 9 — Add Alien (Insert into DB)

In DAO:

```java
public void addAlien(Alien a){
    session.save(a);
}
```

In Controller:

```java
repo.addAlien(a);
```

🌼 PART 10 — Spring MVC + Hibernate = MANY CONFIGS

You saw:

- Many dependencies
- Dispatcher servlet xml
- ViewResolver
- SessionFactory bean
- Transaction Manager
- DataSource
- Packages to scan
- Component-scan
- Annotated classes

This becomes too heavy.

🌼 PART 11 — Spring Boot SOLVES ALL PROBLEMS

Spring Boot says:

👉 “You focus on logic
👉 I will handle configuration.”

So in Spring Boot you need only:

1️⃣ Dependency

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

2️⃣ Dependency

```xml
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
</dependency>
```

Spring Boot will auto configure:

✔️ DataSource
✔️ SessionFactory
✔️ EntityManager
✔️ Hibernate
✔️ Transaction Manager
✔️ Repositories

🌼 PART 12 — application.properties (very simple)

You only need:

```properties
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.datasource.url=jdbc:mysql://localhost:3306/telusko
spring.datasource.username=root
spring.datasource.password=yourpassword
```

That's it.

And Boot will magically start DB connection. 🚀

🌼 PART 13 — Creating Repository (no DAO needed)

Spring Boot + JPA allows this:

```java
public interface AlienRepo extends JpaRepository<Alien, Integer> {

}
```

No need to write:

- getAll
- getById
- save
- delete

Because JpaRepository ALREADY has them:

```java
repo.findAll();
repo.findById(id);
repo.save(alien);
repo.delete(alien);
```

This is why Boot is so easy.

🌼 PART 14 — Using Repo in Controller

```java
@Autowired
AlienRepo repo;
```

Fetch all:

```java
List<Alien> aliens = repo.findAll();
```

Add:

```java
repo.save(alien);
```

Get one:

```java
repo.findById(aid);
```

🌼 PART 15 — Why Boot is Better

Spring MVC	Spring Boot
Many XMLs	Zero XML
Manual configuration	Auto configuration
Many dependencies	Few starter dependencies
Create DAO manually	JPA Repo ready-made
Create SessionFactory manually	Boot auto creates
Many errors	Fewer errors
Slow	Fast

This is why everyone uses **Spring Boot** today.

🌼 Samiksha, tell me what you want next:

1️⃣ Explain Hibernate Query Language (HQL/JPQL) simply

2️⃣ Explain JPARepository custom methods (findByName etc.)

3️⃣ Explain how CRUD works in Spring Boot

4️⃣ Explain Service Layer and clean architecture

5️⃣ Explain MVC flow with diagram

Just say:

👉 **“Explain number 1 next”**
or
👉 **“Explain everything slowly from the beginning”**

I will follow your style.

repository this is the error because it says no default Constructor for entity the thing is when you have this class alien which has a parameterized Constructor it also needs a default Constructor since you are making it entity we forgot that point so let's create create a Constructor here I'll just type it we don't want to write anything in that that's why I'm just specifying this let me restart the application once again go back to the safari and hit refresh and can you see that we got the

04:00:53
output I know that's weird right because nowhere we have written the code for fetching the data we have not implemented the interface which is alien repo and that's the magic the library which we are working with here is is the data jpa now data jpa says say everyone works on C repository right everyone works on CR create read update and delete so when you have those operations what if I can do it for you so spring data jpa says Hey developer you don't have to worry about those operations I

04:01:29
will do it so it has certain classes which implements JP repository and it will take care of all the operations the basic operations and that's why we specify JP repository which will work with alien so all the operations finding uh deleting saving everything you will get just because of this line which is JP repository and that's the beauty so yes there is a class which implements this repository but we are not defining it it is done by Spring framework and that's why I'm in love with spring boot

04:02:01
so just let me just go for a quick recap what we have done we have created this layer which is alien repository you can imagine this is a dow which extends the jpa repository here now jpa repository is responsible to to give you all the basic operations in the application. properties we have specified four properties one of the URL username password and the dialect and that's it we have one more in Alien we have to make sure that this is entity this is at ID and we got a default Constructor now

04:02:31
if you compare this code with the code which we have written here look at the Dow layer code look at the code how many lines are there in fact we just have three methods here what if you have save method what if you want to have remove method or save is there what if you want to have update so this code will go up in fact the idea is the more code you write the more bugs you adding in the software right and that's why spring board is so awesome you write less code and you get less errors so that's it

04:03:01
from this video in which we have talked about spring data GPA so now let's implement the get alien which is fetching only one particular alien and how to add a new alien so we already have those methods here in the controller right we have get alien and then we have ADD alien now first of all we don't need this list anymore let's remove that and here instead of rning the static code which we have done let's return the data from the repository now we don't have to add any method in the

04:03:34
repo because everything is coming from JP repository so what we'll do here is we'll say repo dot now which method we use to fetch that so we have a method which is get so the method name is get one in which you have to pass the ID which is a ID in this case and that's it your job is done it will give you that particular element let's also implement the same thing or maybe let's run this code first so let's relaunch the application and in the Safari let's go back to the homepage say refresh now

03:53:26
compulsory let's do it then so let's go back to the application of properties and here we have to provide the configuration now what config you ask the first one you have to mention the URL in fact first of all we add the dialect so when you work with hnet see behind the scene hibern is running it right so or jpa hnet is running it so we have to specify the dialect here so let's say spring. jpa do properties why I'm typing all this thing when I can do the shortcut so you don't have to type

03:53:57
it you simply have to say hnet do dial that's weird I'm not getting the entire stuff let me just type it here spring. jpa do properties. hybernate D now once you got the dialect you have to specify the dialect name which dialect you actually working with so I'm working with myql but that belongs to org. hybernate do dialect. mysql5 dialect okay so as your dialect changes this will change so if you're using post this becomes post dialect that's the confusion to do now the next

03:54:30
three important the first one is URL now when you work with jdbc you know what we need we need URL we need username and we need password so let's assign that here here so I will say data source. URL the first configuration uh so that is your jdbc colon MySQL colon SL localhost colon 3306 slash database name is teliscope the next one we need is username and password so I will say username so that's for data source. username where is that it's here so username we want is root and data

03:55:06
source. password and in this case is 1 2 3 4 5 6 7 8 that's my database password that's done we have provided the configuration we just want to solve this eror first so let's run this L let's see what happens the moment you relaunch the application now you can see we have not got any error so the only thing we have done is we have added dependencies and we have configured the application. properties Next Step would be we need to create a dow layer then we have to go to Home controller we have to use Dow layer

03:55:39
uh we have to also make sure that your alien class has entity annotation and at ID but that we'll do in the next [Music] video now once we got our dependencies and then we have also configured our application. Properties by specifying this four properties here let's create our Dow layer now if you remember in Spring MVC as well we were supposed to create a alien Dow class but what we will do here is instead of creating a class me create an interface I'll right click here I will say let me get an

03:56:12
interface and we'll name this interface as alien rappo now you might be thinking okay so we got the interface here we'll be writing some methods here you know the Cod operations add alien get alien maybe you can also try removing the alien and then we will create some class which will implement this particular interface that's what we are thinking about let's see that later but what I will do is I will simply extend this class or this interface with another interface named as jpa

repository so

03:56:45
this jpa repository will ask you for two things one is it will ask you for the class which is a in this case and the second one it will ask you for the type of primary key in this case we are using integer let me import the package so you can see we have created the repo here okay so the point is if you got the repo which is your da in fact so once you got the repo you have to create a class which will implement interface we'll do that in some time uh let me just go back to home controller and now I want to

03:57:15
instantiate the repo and the way you do that is very simple you say alien repo uh let's name it as repo itself the way we have done for da and let's s it autowired now one thing which is very important if you want to instantiate this alien repo you need a class and we have not done that till now okay let's keep it for later now once we got the repo we can simply perform certain operations here right we can call those methods we'll do that but before that I want to go back to alien and in Alien we

03:57:47
have to specify that this is an entity right and then we have to also mention the at ID annotation here so that's done let's go back to home controller now once you got the repo let's first try to convert this get aliens here so I don't want the static names so let me remove that so how will you get the aliens it's very simple you simply say repo dot now of course in the interface you can see we don't have or not this one if you see the interface here we don't have any

03:58:16
method but see the magic the moment you say repo dot can you see that how many methods you got in fact we also have a method which is find all find all is a method which we have not declared and we are not even defining it okay so definition part is later but how it is coming we have not declared it now that's where this jpa repository is there so when you say your interface extends jpa repository jpa repository is a interface belongs to the spring data jpa Library which we have added and if you go there you can see it

03:58:55
has so many methods all are declared you can see we have find all we have find all uh we have find all by ID we have list goes on you know we have uh get methods we have remove there are so many methods available here if you can't find methods in this particular interface even this interface extends another interface here some methods belong here there's no method which is available here some methods are available in code cude repository and you can see we have some methods here okay that's fine uh

03:59:25
you can just check them out later so there are methods which are there in JP repository which you are getting in Alien repository you will say okay but all of them are declared right none of them is defined that means we have to Define them now is it a good thing or a bad thing to Define all the methods okay let let's see that so without defining the class what do you think will this work of course not right in Java we have learned if you have not defined stuff of course this will not work let's try

03:59:54
let's relaunch the application and let's see what may go wrong I will say run as in fact I just have to

restart the application so you can see we have not got any error maybe we will get an anomic eror let's go back to application refresh there's no error till now let me fetch all the aliens so I will just go back to the address bar and I will say get aliens I will say enter oh we got the error okay it's not something we have not got the error but then this is not the error because of that alien

04:00:23
repository this is the error because it says no default Constructor for entity the thing is when you have this class alien which has a parameterized Constructor it also needs a default Constructor since you are making it entity we forgot that point so let's create create a Constructor here I'll just type it we don't want to write anything in that that's why I'm just specifying this let me restart the application once again go back to the safari and hit refresh and can you see that we got the

04:00:53
output I know that's weird right because nowhere we have written the code for fetching the data we have not implemented the interface which is alien repo and that's the magic the library which we are working with here is is the data jpa now data jpa says say everyone works on C repository right everyone works on CR create read update and delete so when you have those operations what if I can do it for you so spring data jpa says Hey developer you don't have to worry about those operations I

04:01:29
will do it so it has certain classes which implements JP repository and it will take care of all the operations the basic operations and that's why we specify JP repository which will work with alien so all the operations finding uh deleting saving everything you will get just because of this line which is JP repository and that's the beauty so yes there is a class which implements this repository but we are not defining it it is done by Spring framework and that's why I'm in love with spring boot

04:02:01
so just let me just go for a quick recap what we have done we have created this layer which is alien repository you can imagine this is a dow which extends the jpa repository here now jpa repository is responsible to to give you all the basic operations in the application. properties we have specified four properties one of the URL username password and the dialect and that's it we have one more in Alien we have to make sure that this is entity this is at ID and we got a default Constructor now

04:02:31
if you compare this code with the code which we have written here look at the Dow layer code look at the code how many lines are there in fact we just have three methods here what if you have save method what if you want to have remove method or save is there what if you want to have update so this code will go up in fact the idea is the more code you write the more bugs you adding in the software right and that's why spring board is so awesome you write less code and you get less errors so that's it

04:03:01
from this video in which we have talked about spring data GPA so now let's implement the get alien which is fetching only one particular alien and how to add a new alien so we already have those methods here in the controller right we have get alien and then we have ADD alien now first of all we don't need this list anymore let's remove that and here instead of rning the static code which we have done let's return the data from the repository now we don't have to add any method in the

04:03:34
repo because everything is coming from JP repository so what we'll do here is we'll say repo dot now which method we use to fetch that so we have a method which is get so the method name is get one in which you have to pass the ID which is a ID in this case and that's it your job is done it will give you that particular element let's also implement the same thing or maybe let's run this code first so let's relaunch the application and in the Safari let's go back to the homepage say refresh now

04:04:01
into fetch let's say one3 let's click on submit and you can see we got K that's what we have in database right so this is working and just look at the code we are not doing much of the code but everything is given by Spring data jpa so let's Implement add as well now what do you think what should be done here it's very simple actually you just have to say session not you have to say repo dot the method name okay we have to search for it now what if you don't know the method name so technically it should

04:04:28
be somewhere around save or add or something uh so if you see we don't have any method like with a so let's try save so it saves there's a method name as save we have we have to pass a so now we are trying to save the alien object uh let me repeat we are not doing any commit and roll back part here everything is done by Spring uh so let's relaunch the application in fact in normal spring MVC as well we were using add transactional so we were not doing those uh begin transaction commit

04:04:59
transaction part so let's see in the Safari let's refresh and okay this time we adding it right so let's say 104 and this time we'll add let's say May now if you click on submit uh you can see we got 104 mayure it is displayed in the page but what about database so let's go back to our myql and let's refresh can you see that we got 104 and mayure I guess I have entered the wrong spading for mayure that's fine but we got the data right that's important so that's

04:05:25
how we can add element and that's how we can fetch a particular element in fact you can also try with remove that will be assignment how can you fire the query to remove something now what if you want to fetch something based on some other parameters so what I will do is let me just go back to the index.jsp and here what if you want to fetch not based on the ID but let's say based on the name or maybe a table has a different parameter when you say a ID that's a primary key right so it is very easy to

04:05:56
fetch something with the help of primary key and that's what your repo also suggests in the repo itself you specified the type of the ID you have which is integer but what if you want to fetch based on the name normally what happens is in a table you can have multiple aliens with the same name how can you fetch something with name because if you go back to home controller and if you let's try this here so if I say let's say I want to fetch something and we have this repo the moment you say get you can see it is

04:06:25
asking you for the ID it is not asking you for the name can we do that can we fetch something with the help of name that will see in the next video so let's try to find a particular alien with the help of name now if you go back to your alien repository we know that this extends jpa repository and this

jpa repository by extending some other interface or it has certain methods in it all the methods are related to the ID because that's how you find the element from the database right with the help of

04:06:59
primary key but now we want to fet something not with the help of primary so what we are trying to do is if I search for a particular a ion I'm not passing the aid I'm passing a name so what I'm saying is let me just copy this code let me create another method and this method name would be get alien by name of course you can have any name doesn't matter but let me just go with this name so get alien by name and here as well the request would be get alien by name I know it doesn't make sense to

04:07:30
have in the URL but that's fine uh we can use the same thing but you can have any URL it doesn't matter here okay so it's not that you have to follow the this rule at this point now here instead of having param as int I will be having string and this will be a name right so we are fetching with the help of a name now here with this repo when you say get one unfortunately we cannot pass a name here because a name is not a primary key so what to do so what if I say you can simply remove this part and you can say

04:08:03
repo dot you can see we have a method which is find by ID so when you have have an ID you can find by ID but now you have to say hey I'm not finding by ID I'm finding by name just type it there so you can say find by a name that's it and just pass a name here but you can see we got an error of course right we don't have this method anywhere and this is your own variable even jpa doesn't know that you want a name there okay so what to do now now since this method is not available in jpa what if

04:08:36
you create this method in your own repository let's create that let me just click on this and say create method find by name in the alien repo you can see we got the method and this will return you uh okay now the thing is it is possible that in a database you might be having multiple aliens not one so in this case let me just go with the list of aliens and here as well we need to return the list okay let's go back here and now will this work now of course right if you talk about all these methods which

04:09:08
are available in JP repository if jpa says I will Implement all the methods that's fine because it is its own uh declaration they have their own definition but now this is our declaration right so that means now we have to get a class which implements alien repository to Define this particular method okay let's not do that let's run this code let's see what happens we maybe we'll get some errors let's run this code and let's go back to Safari so this time I'm fetching an

04:09:37
alien not by by ID but by name or but we don't have a form there we also need a form so what I will do is I just copy this code and paste it here this time I will say enter your name and this will be get alien by name is it the same thing we have yeah get in by name let's go back here and anything mention yeah so we have to mention this a name in this case let's go back to the controller just to verify yes we have a name we have a name let's run this now so let's relaunch the application

04:10:08
go back to the homepage refresh okay oh if you still not got the form why oh I'm in the wrong form

my bad let me just copy this and now since we have multile project that's where the confusion starts right so let's go back to index.jsp here and paste so now let's relaunch the application so refresh and you can see we got enter by name so I will enter name naven let's see if that works click on submit oh it worked maybe it will not work with the second one let me just type row okay it worked let me type some

04:10:45
weird name I will say ready we don't have that in the database okay we got empty Le that means it is fetching from database and the amazing thing is it is working and that's the magic of spring jpa or spring data jpa now this thing here it has a special feature and it is known as the query DSL now DSL stands for domain specific language which means uh depend upon what variables you have depend upon what properties you have jpa will provide you the queries but that doesn't means you can use anything you

04:11:17
want you have to follow a particular thing here so example it should start with find by you can't simply say anything else here so it can be get by or find by and then here after that this should be your variable name the first letter has to be Capital you can't say a name or you can't say anything else you can't even say name because we don't have a variable as name the variable name is a name so you have to follow that protocol the first thing is you have to say find by or get by then you

04:11:46
have to mention the variable name you have to also pass the name which you are searching for so that's how we can specify our own stuff but what if you want to have something more let's say if you want to provide descending order you can also do that you can also say order by if you got a list and you can if you want to reverse it you can do that you can say order by a name let's say control space okay but let's verify this once and go back to your Safari I'm fetching by name I will just submit and

04:12:14
you can see at least we got one name but what if if I go back to database let me just insert some more in fact why from database we can do it from here uh let's say it is 106 and let me add naven one more time oh we have how can I say order by naven because we only have one record right my bad let's try so I will say submit and you can see we got two nain here and first is 101 second is 106 so I will say order by a ID a name uh here as well have to change it so I have to say order by Aid let's relaunch and let me

04:12:49
say refresh here they are in sequence let's also say descending order so that's what we do in the SQL right we say order buy and then we also say DC if you want to see in Reverse okay this is what we wanted right so you can see we got a 106 as first and 101 as second so that's how we can play with uh with this in fact we can also apply or and and here so you can say find by a name and you can pass the first parameter you can also say or find by a name with some other name so you can try that so you

04:13:20
can have multiple methods available so that's the beauty of query DSL here so that's great we have query DSL we can have multiple methods here but what if you want to have your own query can we have our own queries here that will see later till this point we have seen how to work with jpa repository now jpa repository provides you certain features right you can create resource read

update delete but all this thing works with the help of primary key but what if you want to fet some data with some other parameter

04:13:57
which is not a primary key in this case so yes you can create those methods just by writing find by and mention the variable name you can also specify is it order by is it live or not like or something like that you can mention all those stuff but what if you don't want to use methods but you want to use the query okay I'm not talking about SQL query here we can do that but what if you want to use jpql what if you want to use hql uh so this JP repository or spring boot data provides you that

04:04:01
into fetch let's say one3 let's click on submit and you can see we got K that's what we have in database right so this is working and just look at the code we are not doing much of the code but everything is given by Spring data jpa so let's Implement add as well now what do you think what should be done here it's very simple actually you just have to say session not you have to say repo dot the method name okay we have to search for it now what if you don't know the method name so technically it should

04:04:28
be somewhere around save or add or something uh so if you see we don't have any method like with a so let's try save so it saves there's a method name as save we have we have to pass a so now we are trying to save the alien object uh let me repeat we are not doing any commit and roll back part here everything is done by Spring uh so let's relaunch the application in fact in normal spring MVC as well we were using add transactional so we were not doing those uh begin transaction commit

04:04:59
transaction part so let's see in the Safari let's refresh and okay this time we adding it right so let's say 104 and this time we'll add let's say May now if you click on submit uh you can see we got 104 mayure it is displayed in the page but what about database so let's go back to our myql and let's refresh can you see that we got 104 and mayure I guess I have entered the wrong spading for mayure that's fine but we got the data right that's important so that's

04:05:25
how we can add element and that's how we can fetch a particular element in fact you can also try with remove that will be assignment how can you fire the query to remove something now what if you want to fetch something based on some other parameters so what I will do is let me just go back to the index.jsp and here what if you want to fetch not based on the ID but let's say based on the name or maybe a table has a different parameter when you say a ID that's a primary key right so it is very easy to

04:05:56
fetch something with the help of primary key and that's what your repo also suggests in the repo itself you specified the type of the ID you have which is integer but what if you want to fetch based on the name normally what happens is in a table you can have multiple aliens with the same name how can you fetch something with name because if you go back to home controller and if you let's try this here so if I say let's say I want to fetch something and we have this repo the moment you say get you can see it is

04:06:25
asking you for the ID it is not asking you for the name can we do that can we fetch something with

the help of name that will see in the next video so let's try to find a particular alien with the help of name now if you go back to your alien repository we know that this extends jpa repository and this jpa repository by extending some other interface or it has certain methods in it all the methods are related to the ID because that's how you find the element from the database right with the help of

04:06:59
primary key but now we want to fet something not with the help of primary so what we are trying to do is if I search for a particular a ion I'm not passing the aid I'm passing a name so what I'm saying is let me just copy this code let me create another method and this method name would be get alien by name of course you can have any name doesn't matter but let me just go with this name so get alien by name and here as well the request would be get alien by name I know it doesn't make sense to

04:07:30
have in the URL but that's fine uh we can use the same thing but you can have any URL it doesn't matter here okay so it's not that you have to follow the this rule at this point now here instead of having param as int I will be having string and this will be a name right so we are fetching with the help of a name now here with this repo when you say get one unfortunately we cannot pass a name here because a name is not a primary key so what to do so what if I say you can simply remove this part and you can say

04:08:03
repo dot you can see we have a method which is find by ID so when you have have an ID you can find by ID but now you have to say hey I'm not finding by ID I'm finding by name just type it there so you can say find by a name that's it and just pass a name here but you can see we got an error of course right we don't have this method anywhere and this is your own variable even jpa doesn't know that you want a name there okay so what to do now now since this method is not available in jpa what if

04:08:36
you create this method in your own repository let's create that let me just click on this and say create method find by name in the alien repo you can see we got the method and this will return you uh okay now the thing is it is possible that in a database you might be having multiple aliens not one so in this case let me just go with the list of aliens and here as well we need to return the list okay let's go back here and now will this work now of course right if you talk about all these methods which

04:09:08
are available in JP repository if jpa says I will Implement all the methods that's fine because it is its own uh declaration they have their own definition but now this is our declaration right so that means now we have to get a class which implements alien repository to Define this particular method okay let's not do that let's run this code let's see what happens we maybe we'll get some errors let's run this code and let's go back to Safari so this time I'm fetching an

04:09:37
alien not by by ID but by name or but we don't have a form there we also need a form so what I will do is I just copy this code and paste it here this time I will say enter your name and this will be get alien by name is it the same thing we have yeah get in by name let's go back here and anything mention yeah so we have to mention this a name in this case let's go back to the controller just to verify yes we have a name we have a name let's run this now so let's relaunch the application

04:10:08
go back to the homepage refresh okay oh if you still not got the form why oh I'm in the wrong form my bad let me just copy this and now since we have multile project that's where the confusion starts right so let's go back to index.jsp here and paste so now let's relaunch the application so refresh and you can see we got enter by name so I will enter name naven let's see if that works click on submit oh it worked maybe it will not work with the second one let me just type row okay it worked let me type some

04:10:45
weird name I will say ready we don't have that in the database okay we got empty Le that means it is fetching from database and the amazing thing is it is working and that's the magic of spring jpa or spring data jpa now this thing here it has a special feature and it is known as the query DSL now DSL stands for domain specific language which means uh depend upon what variables you have depend upon what properties you have jpa will provide you the queries but that doesn't means you can use anything you

04:11:17
want you have to follow a particular thing here so example it should start with find by you can't simply say anything else here so it can be get by or find by and then here after that this should be your variable name the first letter has to be Capital you can't say a name or you can't say anything else you can't even say name because we don't have a variable as name the variable name is a name so you have to follow that protocol the first thing is you have to say find by or get by then you

04:11:46
have to mention the variable name you have to also pass the name which you are searching for so that's how we can specify our own stuff but what if you want to have something more let's say if you want to provide descending order you can also do that you can also say order by if you got a list and you can if you want to reverse it you can do that you can say order by a name let's say control space okay but let's verify this once and go back to your Safari I'm fetching by name I will just submit and

04:12:14
you can see at least we got one name but what if if I go back to database let me just insert some more in fact why from database we can do it from here uh let's say it is 106 and let me add naven one more time oh we have how can I say order by naven because we only have one record right my bad let's try so I will say submit and you can see we got two nain here and first is 101 second is 106 so I will say order by a ID a name uh here as well have to change it so I have to say order by Aid let's relaunch and let me

04:12:49
say refresh here they are in sequence let's also say descending order so that's what we do in the SQL right we say order buy and then we also say DC if you want to see in Reverse okay this is what we wanted right so you can see we got a 106 as first and 101 as second so that's how we can play with uh with this in fact we can also apply or and and here so you can say find by a name and you can pass the first parameter you can also say or find by a name with some other name so you can try that so you

04:13:20
can have multiple methods available so that's the beauty of query DSL here so that's great we have query DSL we can have multiple methods here but what if you want to have your own query can

we have our own queries here that will see later till this point we have seen how to work with jpa repository now jpa repository provides you certain features right you can create resource read update delete but all this thing works with the help of primary key but what if you want to fet some data with some other parameter

04:13:57
which is not a primary key in this case so yes you can create those methods just by writing find by and mention the variable name you can also specify is it order by is it live or not like or something like that you can mention all those stuff but what if you don't want to use methods but you want to use the query okay I'm not talking about SQL query here we can do that but what if you want to use jpql what if you want to use hql uh so this JP repository or spring boot data provides you that

04:14:26
feature so what you can do is uh if you want to use a query it's very simple first of all what I will do is I will just come in this section uh so I want to fetch the alien based on their name but not using this method so I want a different method name here so I will say list of alien and I will have the method name as find just to you know just to show that it is not matching with any of the method which we have done before and then in this I want to accept a particular variable so I will say string

04:14:55
a name the way we have done in the about code okay so once we have done that we have to do one more change if you go back to your home controller here as well you have to mention the method name as find uh so it's not anymore find bar a name order buy something we just have to mention find go back to Alien report and now here this will not work right because we have not specified in the name what you want to fetch in fact what you want to pass and is it aut by or something so in that case you can

04:15:23
specify the query here itself and if you want to specify the query we need to use a query annotation so I'll be saying atate query uh okay you have to make sure that you're coming you're taking it from JP repository now once you got the qu annotation in bracket you can specify the query which you want now in this case I'm using jpql so I'll be saying from Alien so I'm not using SQL where you have to say select Stu from stuff you can simply say from Alien where a name is equal to okay now this uh the

04:15:54
actual value is coming in the method itself right so whatever value you're receiving here from the controller will be passed in the query so in that case you have to use a placeholder now if you are coming from jdbc you know we have used uh what do you say prepared statement where you pass question marks here we don't have to pass question mark we have to use placeholders we can say colon a name right but how do you specify that whatever is coming in the a name will be replaced here or maybe I will say simply

04:16:21
say name just to show the difference so I want to specify this name is a placeholder for this a name and the way you can do that is by using a param annotation you have to make sure that is is coming from uh the repository data repository again and in the bracket you have to mention the name which using which you're fetching it so it's name so whatever is coming in this a name will be assigned to this name at the run time okay that's what uh we have specified here let's verify if it is working let's

04:16:52
relaunch the application go back to our safari and now I'm fetching a particular alien click on submit and it worked can you see that it worked so yes we can use the methods or we can use Query in fact the advantage with query you can pass complex queries okay not just this is simple but try different complex queries here so that's about how can we use Query in jpa So when you say client and server we have two different entities here and normally we imagine that client will be a human normally that's what we do right

04:17:28
so as a human we go to different websites and we access the different resources available online so basically there's a server there now server is responsible to send your data and as a client you consume data now as a human we want data in a proper presented format we don't want our data in XML or Json or a plain text we want our data with a good layout like HML we have images we have Mars we have headers we have all these different amazing designs there so basically server is responsible

04:18:02
to serve you HTML but then now things are changing initially when you say client server we were having machines and humans right so human will be client machine will be server but now we have two applications let's say if you are building two different websites or web resources or web services now one web service want to access data from the second web service now mind you here the first machine is a client the second machine is a server so in this case even the client is a machine now as a client

04:18:32
we don't want HTML right I mean as a machine we don't want HTML we only want data so if one service want to access the second service data it's only about data it's not about the design so let's take an example here so let's say you have a Android app which will be showing you the current weather okay so now the thing is your application will not know the exact weather for this particular area or maybe in the weather in us or maybe in specific area in India so you're not sure exactly the values right

04:19:05
so now here you will request the servers so we have different services available so you can send a request to them by saying hey I want to know the current temperature I want to know the current humity and This Server will send you the data not design only data because this app has a design inbuild So when you say you are sending data so how it will request so we only want data but then you will simply say okay we can do that so we can use Sur JSP we can send request will return data in fact we can

04:19:33
do that in normal spring MC as well but we have second issue here the second thing is when we talk about Sate or JSP we normally perform actions if you remember when we create forms in forms we say action and we specify the name now that is your URL and you also specify the action which you want to take so one action will be having one URL but then what if you don't want to depend upon action you want to depend upon noun okay something like this so let's say you're going to a website you

04:20:02
have a website which provides you a lot of resources it provides you all the employee Nam it provides you all the account details and different stuff now if you want employee details instead of

saying ww. abc.com query is equal to employ you can simply say ww. abc.com employ now which makes more sense having Q equal to employee or uh Slash employe the employees right so you're specifically mentioning that you are going for a noun here you want employes so that's what we are going for so we

04:20:33
are going for a good URI with with the help of nouns and and we are going for only data now this data will be basically the object State okay so let's say on the server we have lot of objects and you want to send data in fact the Java we do that so every object will be having data and this is represented by some pojo class so if you just want to send the data which is the current object state so you're sending the state of the object and that's why this can be done with the help of representational....give important and teach me like this only in very very simple

Samiksha... 🤍
I read everything you pasted **VERY carefully** — it's a long transcript from a Spring Boot + Spring Data JPA tutorial.

You said:

👉 **"Give only the IMPORTANT parts and teach me like this only, very very simple."**

So now, I will explain the **EXACT part you highlighted**
(**No default constructor error + how JPA Repository does magic**)
in **super simple beginner language**, like the trainer, but easier.

🌼 PART 1 — Why Spring showed the error:

❌ "No default constructor for entity"

Very simple:

🎒 Think of your entity (Alien class) like a school form.

Spring wants to **create an empty object first** and then fill values inside.

But your class had only:

```java
public Alien(int aid, String aname) {
    ...
}
```

This is **parameterized constructor only**.

So when Spring tries:

```java
new Alien();
```

It fails ❌
Because this constructor does **not exist**.

✔️ FIX:

Add a **default constructor**:

```java
public Alien() {
}
```

This is enough for Spring & JPA to be happy.

🌼 PART 2 — MAGIC of Spring Data JPA

After fixing the constructor, Spring Boot suddenly works.

And you asked:

👉 "How? We did NOT write fetch code. We did NOT implement interface. But still we get output?"

❤️ Simple answer:

Because Spring Data JPA writes everything FOR YOU.

You create this interface:

```java
public interface AlienRepo extends JpaRepository<Alien, Integer> {
}
```

And BOOM! 💥
You automatically get:

✔️ findAll()
✔️ findById()
✔️ save()
✔️ delete()
✔️ deleteById()
✔️ count()
✔️ getOne()

You didn't write these methods.
You didn't implement anything.

But Spring Data JPA provides these:

```arduino
Already implemented in background.
```

✔️ You only call:

```java
repo.findAll();
repo.getOne(id);
repo.save(alien);
```

And everything works.

🌼 PART 3 — Why JPAT Repository gives so many methods?

Because:

📌 JPARepository → extends PagingAndSortingRepository

📌 PagingAndSortingRepository → extends CrudRepository

📌 CrudRepository → contains all CRUD methods

So your interface inherits all methods automatically.

That's why you get 20+ methods without writing any code.

🌼 PART 4 — Why Spring Boot makes everything easy?

Earlier in Spring MVC you wrote:

- DataSource config
- SessionFactory config
- Hibernate properties
- Transaction manager
- DAO class
- Implement interface
- Add XML config

Now in Spring Boot:

Only 4 lines in `application.properties`:

```ini
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.datasource.url=jdbc:mysql://localhost:3306/telusko
spring.datasource.username=root
spring.datasource.password=12345678
```

And:

✔️ Boot creates DataSource
✔️ Boot creates SessionFactory
✔️ Boot creates EntityManager
✔️ Boot handles transactions
✔️ Boot implements repository

Magic ✨

🌼 PART 5 — FETCH one Alien using ID

In controller:

```java
repo.getOne(id);
```

This internally runs:

```csharp
select * from alien where aid = ?
```

You don't write this query → JPA writes it.

🌼 PART 6 — ADD Alien

Just call:

```java
repo.save(alien);
```

✔️ For new object → INSERT

✔️ If ID exists → UPDATE

All automatic 💛

🌼 PART 7 — FIND BY NAME (Not Primary Key)

You wrote:

```java
List<Alien> findByAname(String aname);
```

Spring automatically converts it into a query:

```
csharp

select * from alien where aname = ?
```

Why does it work?

Because of **Query DSL**

Rules:

- Method must start with
  ✔️ findBy
  ✔️ getBy
- The variable name must match the entity field
  ✔️ Aname → Aname
  ✔️ Not name, not aname1, not fullname
- First letter after "By" must be Capital

Examples:

✔️ findByAname
✔️ getByAname
✔️ findByAnameOrderByAidDesc
✔️ findByAnameAndAid
✔️ findByAnameOrAid

All automatically work 👌

🌼 PART 8 — ORDER BY easily

Example:

```
java

List<Alien> findByAnameOrderByAidDesc(String aname);
```

This becomes:

```
pgsql

select * from alien
where aname = ?
order by aid desc
```

You didn't write query → JPA wrote it.

🌼 PART 9 — Using your own JPQL Query

If you don't want DSL method, you can write your own JPQL:

```java
@Query("from Alien where aname = :name")
List<Alien> find(@Param("name") String aname);
```

✔️ "from Alien" = JPQL
✔️ No need to write table name
✔️ No need to write SELECT
✔️ Cleaner + Object-based

🌼 PART 10 — Why do we need @Entity + @Id?

Because JPA must know:

- This class represents a table → @Entity
- Which field is primary key → @Id

If @Id is missing → error
If default constructor missing → error

🌼 PART 11 — SUMMARY IN ULTRA SIMPLE LANGUAGE

💙 WITHOUT Spring Boot

You write:
❌ DAO class
❌ find method
❌ save method
❌ delete method
❌ sessionFactory
❌ hibernate.cfg
❌ transaction open/commit
❌ 100+ lines setup

💚 WITH Spring Boot

You write:
✔️ **`@Entity`**
✔️ **`@Id`**
✔️ **`AlienRepo extends JpaRepository`**
✔️ **`repo.save()`**
✔️ **`repo.findAll()`**

Everything else:

✨ Spring Boot does for you
✨ Spring Data JPA implements methods

✨ Hibernate runs queries
✨ Spring manages transactions

You write 10 lines → Boot writes 1000 lines internally.

🌼 Samiksha, your next step?

Tell me what you want next explained **in this simple style**:

1️⃣ JPQL & HQL fully

2️⃣ RestController (Nouns, JSON, API)

3️⃣ Service layer (Why needed)

4️⃣ Entity relationships (OneToMany, etc.)

5️⃣ How Spring Boot handles transactions

6️⃣ How Hibernate creates SQL queries internally

Just say:
👉 **Explain number __ next**

I'll teach you again in **super easy language** ❤️

04:21:02
state and we are transferring so it will be transfer and in short we call it as rest okay so that's why we have this very famous API or the service named as restful web service or API so this is a API which will give you data that will be representational State okay that will be we are sending to the client in fact client can also send data to the server the amazing thing is when you talk about MVC when you talk about Ser GSP we maintain session so server is responsible to maintain some data but

04:21:31
that's not the case with rest in rest scenario what we do is uh every request is stateless okay so even server don't know that this client has sent the request before so that means every request from a client will have all the data you can't simply Say Hey I was there last week you can't do that because server don't know you every request will be stateless so in that case you know if you want to let's say if you want to maintain login if you want to maintain some data that you want

04:21:58
to inform server hey I was there last week you have to send a token okay so that means when you have sent the first request to the server server will be given giving you some token so next time when you go to the server you have to carry that token so these are the some things which are important when you talk about Rest full web service so the first point is we are not working with actions we are working with nouns okay so you simply saying ww. abc.com employees slash aliens

SL laptops or even if you want to pick up a

04:22:27
specific laptop let's say the laptop of Lenovo so you can simply say/ laptops SL Lenovo it will only give you Lenovo laptops uh second everything will be stateless so you can't simply maintain session on the server and we are talking about the state of the object here but how will you implement that so we are not going to use Q equal to something so we are not going to pass parameters in the address bar and the client will not be a normal browser oh there's one more important thing when you say rest it

04:22:56
uses the inbuild methods of the sttp protocol so we have certain methods right so we have get post put and delete so we'll be using this method so if you want to create a new data on the server so normally we call them resources data will be resources so if you want to create a new resource on the server you will say post if you want to update the existing resource you'll be using put if you want to fetch the resource you will say get so we can use all this normal HTP methods in rest API but how exactly

04:23:26
we can do that that we'll see in the next video once we know the theory of rest let's try to implement it but hold on we have a issue here when you say you have a normal web application so we have a server and then we have a client so normally when we imagine client we are talking about humans but on the other side when you talk about rest it's more about a client server but then here the server will provide you some content but not a design or not in a format of HTML So when you say a client it will be a

04:24:00
machine or it will be an application now this application we have to build it maybe Android maybe a website but then since we don't want to do all those stuff we don't want to do something extra let's use a client which is provided by someone else the client which we are going to use is specially used for rest testing or rest API use the client name is Postman in fact we have so many rest clients available but Postman is quite famous I will say Postman initially I was using it with as

04:14:26
feature so what you can do is uh if you want to use a query it's very simple first of all what I will do is I will just come in this section uh so I want to fetch the alien based on their name but not using this method so I want a different method name here so I will say list of alien and I will have the method name as find just to you know just to show that it is not matching with any of the method which we have done before and then in this I want to accept a particular variable so I will say string

04:14:55
a name the way we have done in the about code okay so once we have done that we have to do one more change if you go back to your home controller here as well you have to mention the method name as find uh so it's not anymore find bar a name order buy something we just have to mention find go back to Alien report and now here this will not work right because we have not specified in the name what you want to fetch in fact what you want to pass and is it aut by or something so in that case you can

04:15:23
specify the query here itself and if you want to specify the query we need to use a query annotation so I'll be saying atate query uh okay you have to make sure that you're coming you're

taking it from JP repository now once you got the qu annotation in bracket you can specify the query which you want now in this case I'm using jpql so I'll be saying from Alien so I'm not using SQL where you have to say select Stu from stuff you can simply say from Alien where a name is equal to okay now this uh the

04:15:54
actual value is coming in the method itself right so whatever value you're receiving here from the controller will be passed in the query so in that case you have to use a placeholder now if you are coming from jdbc you know we have used uh what do you say prepared statement where you pass question marks here we don't have to pass question mark we have to use placeholders we can say colon a name right but how do you specify that whatever is coming in the a name will be replaced here or maybe I will say simply

04:16:21
say name just to show the difference so I want to specify this name is a placeholder for this a name and the way you can do that is by using a param annotation you have to make sure that is is coming from uh the repository data repository again and in the bracket you have to mention the name which using which you're fetching it so it's name so whatever is coming in this a name will be assigned to this name at the run time okay that's what uh we have specified here let's verify if it is working let's

04:16:52
relaunch the application go back to our safari and now I'm fetching a particular alien click on submit and it worked can you see that it worked so yes we can use the methods or we can use Query in fact the advantage with query you can pass complex queries okay not just this is simple but try different complex queries here so that's about how can we use Query in jpa So when you say client and server we have two different entities here and normally we imagine that client will be a human normally that's what we do right

04:17:28
so as a human we go to different websites and we access the different resources available online so basically there's a server there now server is responsible to send your data and as a client you consume data now as a human we want data in a proper presented format we don't want our data in XML or Json or a plain text we want our data with a good layout like HML we have images we have Mars we have headers we have all these different amazing designs there so basically server is responsible

04:18:02
to serve you HTML but then now things are changing initially when you say client server we were having machines and humans right so human will be client machine will be server but now we have two applications let's say if you are building two different websites or web resources or web services now one web service want to access data from the second web service now mind you here the first machine is a client the second machine is a server so in this case even the client is a machine now as a client

04:18:32
we don't want HTML right I mean as a machine we don't want HTML we only want data so if one service want to access the second service data it's only about data it's not about the design so let's take an example here so let's say you have a Android app which will be showing you the current weather okay so now the thing is your application will not know the exact weather for this particular area or maybe in the weather in us or maybe in specific area in India so you're not sure

exactly the values right

04:19:05
so now here you will request the servers so we have different services available so you can send a request to them by saying hey I want to know the current temperature I want to know the current humity and This Server will send you the data not design only data because this app has a design inbuild So when you say you are sending data so how it will request so we only want data but then you will simply say okay we can do that so we can use Sur JSP we can send request will return data in fact we can

04:19:33
do that in normal spring MC as well but we have second issue here the second thing is when we talk about Sate or JSP we normally perform actions if you remember when we create forms in forms we say action and we specify the name now that is your URL and you also specify the action which you want to take so one action will be having one URL but then what if you don't want to depend upon action you want to depend upon noun okay something like this so let's say you're going to a website you

04:20:02
have a website which provides you a lot of resources it provides you all the employee Nam it provides you all the account details and different stuff now if you want employee details instead of saying ww. abc.com query is equal to employ you can simply say ww. abc.com employ now which makes more sense having Q equal to employee or uh Slash employe the employees right so you're specifically mentioning that you are going for a noun here you want employes so that's what we are going for so we

04:20:33
are going for a good URI with with the help of nouns and and we are going for only data now this data will be basically the object State okay so let's say on the server we have lot of objects and you want to send data in fact the Java we do that so every object will be having data and this is represented by some pojo class so if you just want to send the data which is the current object state so you're sending the state of the object and that's why this can be done with the help of representational

04:21:02
state and we are transferring so it will be transfer and in short we call it as rest okay so that's why we have this very famous API or the service named as restful web service or API so this is a API which will give you data that will be representational State okay that will be we are sending to the client in fact client can also send data to the server the amazing thing is when you talk about MVC when you talk about Ser GSP we maintain session so server is responsible to maintain some data but

04:21:31
that's not the case with rest in rest scenario what we do is uh every request is stateless okay so even server don't know that this client has sent the request before so that means every request from a client will have all the data you can't simply Say Hey I was there last week you can't do that because server don't know you every request will be stateless so in that case you know if you want to let's say if you want to maintain login if you want to maintain some data that you want

04:21:58
to inform server hey I was there last week you have to send a token okay so that means when you

have sent the first request to the server server will be given giving you some token so next time when you go to the server you have to carry that token so these are the some things which are important when you talk about Rest full web service so the first point is we are not working with actions we are working with nouns okay so you simply saying ww. abc.com employees slash aliens SL laptops or even if you want to pick up a

04:22:27
specific laptop let's say the laptop of Lenovo so you can simply say/ laptops SL Lenovo it will only give you Lenovo laptops uh second everything will be stateless so you can't simply maintain session on the server and we are talking about the state of the object here but how will you implement that so we are not going to use Q equal to something so we are not going to pass parameters in the address bar and the client will not be a normal browser oh there's one more important thing when you say rest it

04:22:56
uses the inbuild methods of the sttp protocol so we have certain methods right so we have get post put and delete so we'll be using this method so if you want to create a new data on the server so normally we call them resources data will be resources so if you want to create a new resource on the server you will say post if you want to update the existing resource you'll be using put if you want to fetch the resource you will say get so we can use all this normal HTP methods in rest API but how exactly

04:23:26
we can do that that we'll see in the next video once we know the theory of rest let's try to implement it but hold on we have a issue here when you say you have a normal web application so we have a server and then we have a client so normally when we imagine client we are talking about humans but on the other side when you talk about rest it's more about a client server but then here the server will provide you some content but not a design or not in a format of HTML So when you say a client it will be a

04:24:00
machine or it will be an application now this application we have to build it maybe Android maybe a website but then since we don't want to do all those stuff we don't want to do something extra let's use a client which is provided by someone else the client which we are going to use is specially used for rest testing or rest API use the client name is Postman in fact we have so many rest clients available but Postman is quite famous I will say Postman initially I was using it with as

04:24:30
a extension for a chrome you can see we still have it I guess it is deprecated uh so they recommend you to to use the application so it doesn't matter is it Windows or Mac it still works on all the OS available so if you click on this so you can see you will get an option of download so depending upon which OS you're working with you can select it so click on download and it a simple installation say next next and your job will be done now in this machine I already have Postman with me so I'll be

04:24:58
using it so it will take some time to download so let's not use that let's use the existing version so I'm launching the postman which I have and it will ask me for login okay I will simply sign up with Google or sign in with Google so I'm starting my Postman here okay so it says the Chrome app is being dicated okay I know that close that and close this one now you can see this is how your Postman looks like I guess the I'm still using the Chrome extension but that's fine I will just install

the new

04:25:30
one for the next video but as of now this will work the UI is same for everything so you can see see we have a history here for all your uis and this is where you will fire a particular request time being let me request google.com and let's see what happens so I will say google.com enter so I'm sending a get request to Google send okay so you can see we got the data here normally this is your HTML page but then when you say rest API you will be getting your data in XML or Json so this is where you will see your XML or Json

04:26:03
we have an option here but then we are not returning the XML or Json that's why we will not be seeing that so this is your get request if you want to work with post it's very easy you can click on post specify the URL or URI here and click on send if you want to use put for updating data you can also use delete for removing the content if you want to send data with some headers you can do that here uh if you want to send actual data you have to put that do that in body we have certain options here again

04:26:30
we'll be seeing that everything one by one so I hope you will be installing Postman for after this and then you'll be doing a proper setup so that you can be ready for the next video now once we have our Postman ready let's change the code now till this point we have worked with normal spring MVC where where we were sending data we were fetching data but not just data we were fetching the entire page that's what we do in normal MVC right and all the requests were actions so we were

04:26:59
sending post request we were sending get request but here we let's use noun so what I will do is I will open my Postman and here we'll send a get request now this time I want to fetch all the aliens now if you see the earlier one if you go back to your code and if you open index.jsp what we are doing is we are sending the we are saying submit and the action is ADD alien so in the address bar it was add alien in fact to fetch all the aliens we were using get aliens but why you have to say get aliens which

04:27:28
sounds like a verb here we want to use noun example if I go back here I will simply say Local Host colon 8080 that's the protocol we are using slash so instead of saying get aliens I will simply say aliens and here you can simply say send and this should give you data now here you can see we don't have any mapping there which will handle aliens and that's why we got an error so the error is it says 44 not found okay that's great we got the error and if you see the we got the error in Json format

04:27:59
that's weird I know uh okay let's let's go back to the application now here what I want to do is now first of all we don't need the t. JSP anymore in fact we don't even need any JSP because we are not using views here we are simply working with Json or XML so let's go back here and now in the controller I want something which will handle my aliens mapping now in fact I can create a new controller or we can change the existing controller now I'm quite famous when it comes to changing the existing

04:28:27

code so let's do that so I will go back here this is the same thing we want right we want this particular method so I will just copy it uh and I will say paste here itself or maybe it will make much more sense if you create a new file right let's do that so here let me create one more controller I will say new and let's name this as a alien controller of course you can have a different name but since you working with alien let's do that so I will say alien controller okay nothing much I

04:28:54
want it to be in the same package that's fine click on finish now you can see we got our alien controller and in this I want that particular method which is the mapping so we got this method right here and then I want this to be a controller so let's do that so I will simply say controller nothing fancy right now we got our class let me put the package first okay but don't you think it should not be get alien should be aliens so I will say aliens here okay and then we have one more we don't need model

04:29:22
anymore because when you say model that means we are working with session we are sending JSP here we are not sending JSP so we don't even need a model but we want to fetch data right and we have done that before we wanted repo as well uh if you remember we also need a alien repo which will fetch data from database let's copy that as well okay so once you got your alien repo let's fetch data so I will simply say report dot we want to fetch all the aliens I will say find all as simple as that but this will return

04:29:52
you the list of aliens now we have a choice you can save that in the list or you can simply write returns report. findall that's your choice I will go step by step I will say least aliens let's fetch aliens here so control space and control space for this as well okay once you got all the aliens here we are not supposed to specify the page name because we're not using JSP so let's specify aliens but will this work think about this this aliens this aliens is a list object we are sending data to

04:30:26
string we don't want that right so what I will do here is I will convert this into two string let's try let's see what happens now once we got this let's run this application and let's see what happen so let's relaunch the application here and go back to postman and here you will send request here and we got an error can you see that it says again not found that's weird we have a mapping for aliens right but look at this output it is giving you the message which is all

04:30:52
the aliens that's weird we got all the aliens but still it says error the thing is that's how your spring m is built right spring MVC says whatever you return I will search the JSP with this name because that's how you we have configured the application if you see our application. properties here also we have said that whatever name you will send I will simply add JSP and we'll search that name in views so whatever you're sending here is actually your JSP name we have to mention hey this is not

04:31:22
a JSP name which I'm sending I'm sending actual data the way you can specify that you are sending actual data is by using another annotation which is response body so when you say response body here it will send that data as a data not GSP name let's relaunch the application go back to postman and here you will say send can you see that we got our data but with small error okay it's not actually the error but we wanted Json right we wanted XML but we got a plain text this is not Jason we want data and

04:31:55
Jason format if you want to achieve that if you want to send data and Jason format don't set two string because when you say two string it will convert your data into string we don't want that so we'll send aliens and here as well will not send string let's return the list as it is now you might be wondering will it work as is yes it will work let's try let's relaunch the application and here if I say send you can see that we got our data in Json format that's what we

04:32:20
wanted right we got Json so that's how we can have a get request now what about post request what if you want to fetch one particular alien that we'll see in the next video so we have done with the first step of rest where we were able to fetch all the aliens and that to with a URI so if you want to see this is a URI which we are using here so we are saying local colon 880 alens so when I say all the aliens when I click on send it will send a get request to the server and from a server side we are returning all the

04:32:57
aliens but that to a decent format the amazing thing is we are not using Json anywhere I mean as a programmer we are not converting our Java list into ad Json and that's where your spring boot provides you this amazing feature so if you can see your M dependencies we have a converter so you can see we already have a converter here which is Jackson so Jackson is responsible to convert your Java data into Json now since it is inbu so we don't have to change the things by ourself it is that easy now

04:33:30
what if you want to return XML can we do that with Jackson that will one question but then at least the Json is working and that is happening with the help of Jackson here if you just go to Google and if you search for jacksson so we are searching for Jackson documentation and that's your official website of it so they have it on GitHub let's go there okay so Jackson basically convert your Java data into Json so you can see it has Json Passa in that we have Jason generator so Jason using Jason generator

04:34:02
you can generate Json files and vice versa as well so what from a client side you receive a Json so it will convert that data into Java object that's how you can use Jackson and in fact you don't have to add in the project will be done by Spring boot now once we know how to fetch all the values in J format and that with the help of Jackson now I want to fetch one particular record normally what we do is in rest API so when we create our uis what what makes much more sense is let's say I want to fetch one particular alien

04:24:30
a extension for a chrome you can see we still have it I guess it is deprecated uh so they recommend you to to use the application so it doesn't matter is it Windows or Mac it still works on all the OS available so if you click on this so you can see you will get an option of download so depending upon which OS you're working with you can select it so click on download and it a simple installation say next next and your job will be done now in this machine I already have Postman with me so I'll be

04:24:58
using it so it will take some time to download so let's not use that let's use the existing version so I'm launching the postman which I have and it will ask me for login okay I will simply sign up with Google or sign in with Google so I'm starting my Postman here okay so it says the Chrome app is

being dicated okay I know that close that and close this one now you can see this is how your Postman looks like I guess the I'm still using the Chrome extension but that's fine I will just install the new

04:25:30
one for the next video but as of now this will work the UI is same for everything so you can see see we have a history here for all your uis and this is where you will fire a particular request time being let me request google.com and let's see what happens so I will say google.com enter so I'm sending a get request to Google send okay so you can see we got the data here normally this is your HTML page but then when you say rest API you will be getting your data in XML or Json so this is where you will see your XML or Json

04:26:03
we have an option here but then we are not returning the XML or Json that's why we will not be seeing that so this is your get request if you want to work with post it's very easy you can click on post specify the URL or URI here and click on send if you want to use put for updating data you can also use delete for removing the content if you want to send data with some headers you can do that here uh if you want to send actual data you have to put that do that in body we have certain options here again

04:26:30
we'll be seeing that everything one by one so I hope you will be installing Postman for after this and then you'll be doing a proper setup so that you can be ready for the next video now once we have our Postman ready let's change the code now till this point we have worked with normal spring MVC where where we were sending data we were fetching data but not just data we were fetching the entire page that's what we do in normal MVC right and all the requests were actions so we were

04:26:59
sending post request we were sending get request but here we let's use noun so what I will do is I will open my Postman and here we'll send a get request now this time I want to fetch all the aliens now if you see the earlier one if you go back to your code and if you open index.jsp what we are doing is we are sending the we are saying submit and the action is ADD alien so in the address bar it was add alien in fact to fetch all the aliens we were using get aliens but why you have to say get aliens which

04:27:28
sounds like a verb here we want to use noun example if I go back here I will simply say Local Host colon 8080 that's the protocol we are using slash so instead of saying get aliens I will simply say aliens and here you can simply say send and this should give you data now here you can see we don't have any mapping there which will handle aliens and that's why we got an error so the error is it says 44 not found okay that's great we got the error and if you see the we got the error in Json format

04:27:59
that's weird I know uh okay let's let's go back to the application now here what I want to do is now first of all we don't need the t. JSP anymore in fact we don't even need any JSP because we are not using views here we are simply working with Json or XML so let's go back here and now in the controller I want something which will handle my aliens mapping now in fact I can create a new controller or we can change the existing controller now I'm quite famous when it comes to changing the existing

04:28:27
code so let's do that so I will go back here this is the same thing we want right we want this particular method so I will just copy it uh and I will say paste here itself or maybe it will make much more sense if you create a new file right let's do that so here let me create one more controller I will say new and let's name this as a alien controller of course you can have a different name but since you working with alien let's do that so I will say alien controller okay nothing much I

04:28:54
want it to be in the same package that's fine click on finish now you can see we got our alien controller and in this I want that particular method which is the mapping so we got this method right here and then I want this to be a controller so let's do that so I will simply say controller nothing fancy right now we got our class let me put the package first okay but don't you think it should not be get alien should be aliens so I will say aliens here okay and then we have one more we don't need model

04:29:22
anymore because when you say model that means we are working with session we are sending JSP here we are not sending JSP so we don't even need a model but we want to fetch data right and we have done that before we wanted repo as well uh if you remember we also need a alien repo which will fetch data from database let's copy that as well okay so once you got your alien repo let's fetch data so I will simply say report dot we want to fetch all the aliens I will say find all as simple as that but this will return

04:29:52
you the list of aliens now we have a choice you can save that in the list or you can simply write returns report. findall that's your choice I will go step by step I will say least aliens let's fetch aliens here so control space and control space for this as well okay once you got all the aliens here we are not supposed to specify the page name because we're not using JSP so let's specify aliens but will this work think about this this aliens this aliens is a list object we are sending data to

04:30:26
string we don't want that right so what I will do here is I will convert this into two string let's try let's see what happens now once we got this let's run this application and let's see what happen so let's relaunch the application here and go back to postman and here you will send request here and we got an error can you see that it says again not found that's weird we have a mapping for aliens right but look at this output it is giving you the message which is all

04:30:52
the aliens that's weird we got all the aliens but still it says error the thing is that's how your spring m is built right spring MVC says whatever you return I will search the JSP with this name because that's how you we have configured the application if you see our application. properties here also we have said that whatever name you will send I will simply add JSP and we'll search that name in views so whatever you're sending here is actually your JSP name we have to mention hey this is not

04:31:22
a JSP name which I'm sending I'm sending actual data the way you can specify that you are sending actual data is by using another annotation which is response body so when you say response body here it will send that data as a data not GSP name let's relaunch the application go back to postman and here you will say send can you see that we got our data but with small error okay it's not actually the error but we wanted Json right we wanted XML but we got a plain text this is not

Jason we want data and

04:31:55
Jason format if you want to achieve that if you want to send data and Jason format don't set two string because when you say two string it will convert your data into string we don't want that so we'll send aliens and here as well will not send string let's return the list as it is now you might be wondering will it work as is yes it will work let's try let's relaunch the application and here if I say send you can see that we got our data in Json format that's what we

04:32:20
wanted right we got Json so that's how we can have a get request now what about post request what if you want to fetch one particular alien that we'll see in the next video so we have done with the first step of rest where we were able to fetch all the aliens and that to with a URI so if you want to see this is a URI which we are using here so we are saying local colon 880 alens so when I say all the aliens when I click on send it will send a get request to the server and from a server side we are returning all the

04:32:57
aliens but that to a decent format the amazing thing is we are not using Json anywhere I mean as a programmer we are not converting our Java list into ad Json and that's where your spring boot provides you this amazing feature so if you can see your M dependencies we have a converter so you can see we already have a converter here which is Jackson so Jackson is responsible to convert your Java data into Json now since it is inbu so we don't have to change the things by ourself it is that easy now

04:33:30
what if you want to return XML can we do that with Jackson that will one question but then at least the Json is working and that is happening with the help of Jackson here if you just go to Google and if you search for jacksson so we are searching for Jackson documentation and that's your official website of it so they have it on GitHub let's go there okay so Jackson basically convert your Java data into Json so you can see it has Json Passa in that we have Jason generator so Jason using Jason generator

04:34:02
you can generate Json files and vice versa as well so what from a client side you receive a Json so it will convert that data into Java object that's how you can use Jackson and in fact you don't have to add in the project will be done by Spring boot now once we know how to fetch all the values in J format and that with the help of Jackson now I want to fetch one particular record normally what we do is in rest API so when we create our uis what what makes much more sense is let's say I want to fetch one particular alien

04:34:39
here which is mayure if you want to fetch mayure we will be saying 104 and the way you do that is by saying first of all this is your actual website in fact you we can replace this with the website name then you will instead of saying aliens you will say alien because you're fetching one particular alien but now you will also pass the AI D which is 104 so instead of saying Q is equal to 104 this makes much more sense to have a u like alien sl14 you know this is more like fetching a particular resource from

04:35:11
a computer so if if you have a Windows machine or any other machine you know when you go to

any particular folder inside that folder if you go to particular folder and then if you search for a file the entire path will be something like ABC pqr SL maybe text.txt that's how you have the path right because it shows the path of the file or the source in the same way we have this here so when you click on send you can see this is not working yet because we don't have any request mapping which

04:35:39
will accept alien sl14 now if you want to achieve that we have to create a method here so we'll say public now this method will not be rning the least right it will return only one particular alien and let's have the name as alien now this alien will accept a particular Aid so I will say a ID here now this has to be mapped so I have to say get mapping since we are using get here and then we'll be having mapping like alien but then we have to also mention 104 here right because that's how you are mapping

04:36:08
so we have to say 104 and here we have to return the alien so the way you can do that is by saying alien I want to fetch first fetch the alien alien equal to using repo so you can say repo do get one so get one is a method which belongs to JP repository which will give you one particular alien and let's pass a ID here and I guess our job is done we can simply return alien but do you think it will work what what is your first reaction now see it makes sense because whatever we have done is almost same with this method we are

04:36:42
missing one thing we have to also say request response body before the method but we have one problem so this might work for 104 what if you have 1 Z3 what if you have 101 okay we have to make it variable right you can't simply say 104 here because it will only work for 104 in fact you know let's try this for 104 ons and let's not accept or maybe I have to accept it so it will give you because we are expecting something here and we are not passing it let's see what data you get so let's go back to the postman and

04:37:12
let's say hit and you can see we got an error it is your status 500 and it says optional a parameter a ID is present in the parameter but it cannot be translated to a null because we are not passing the values so path is there's no 44 this time but then we have to assign the Val to here right so this 104 need to be assigned to Aid so what we have to do is since this 104 will change let's put that in a cudly bracket so you have to say a ID so that's how we work so if you think the URL will be different it

04:37:46
will be alien for sure slash it will be some number we are not sure about the number so we can simply put that in cly brackets a ID and here we have to use an annotation which is path variable now let's restart the application and let's see what happens let's go back here click on send and we got an error okay okay I guess we have missed something here so we have to also mention what is the name which you're trying to map here so you can see I'm mapping this a ID with this particular Aid so we have to

04:38:15
use path variable Aid here and now let's relaunch the application I hope that was the issue so let's click on send and okay we are still getting same eror it says type definition error so let's try find by ID because maybe that there's a conversion issue with get one okay so find by ID gives you a optional aliens that's what happens in Java 8 you know in Java 8 just to make sure that you don't

get null values we get an optional value there so the only thing is you can do here is we can say to uh we can say

04:38:46
or else and let's return the blank alien object so that we can say we have not received we don't have that alien in the database so we can say new alien and we can have a value which is let's say zero comma double quotes Okay so this should work now I hope let's go back there let's send Okay so is working so that there was something wrong with get one okay so that's how you can do it so we can say find by ID what if this ID is not available in database and that's where you can pass blank value so what

04:39:15
I've done is if I go back to my hit and let's say if I search for one Z we don't have this value database so if you click on send you can see you will get empty data and you can say you know we have not found it so that's how you can work with one particular data so till this point we know how to fetch all the aliens and then we can also fetch one particular alien and for that we have defined two methods here one is get aliens which will give you all the aliens and the way you can do that is by

04:39:43
saying get mapping which is very important but then if you can see we are also mentioning one more annotation here which is response body not just here here as well because both these methods will be working with rest I mean they will be returning the data in just format don't you think in future as well we'll be having some more methods which will be let's say fetch one particular alien based on some other parameter or maybe you know you might be having five to six method in the same class now when

04:40:12
you know that all these methods are actually rest methods and which will return Json or XML what makes much more sense is to not write it everywhere it should be common right we can mention on the class LEL itself that whatever you return will be Json or XML and the way you can do that is by changing this controller annotation So when you say controller it will behave like MVC controller but but you can mention that hey this is not a normal MVC this is actually rest and the way you can do that is by simply mentioning a new type

04:40:41
of controller which is a rest controller so we can say control space let's import the package and when you jump to this controller so you can see it is actually based on controller but uh it's more for response body so this is a controller also response body let's restart the project and let's hit the query once again just to see if it is working let's click on send and you can see it is working so whenever you are sure that you want to build a controller but that will be a rest controller always use

04:41:09
rest control instead of using response body on every method but yes let's say if you have five methods out of which you have three methods as rest two methods are normal MVC uh then you don't have a choice you have to go for normal controller and response body on those methods which returns Json XML okay so everything is working but now what if you want to send the data because till this point we have seen how to fetch data but now I want to send data so what we will be doing is if you see database

04:41:38

we have four records here right we have four in fact we have five 106 is also there n so we have five records I want to insert a new alien and that to in from Postman so we don't have a form officially we don't have HTML page where you can send data we have to do that from this particular Postman instead of saying get we'll be using post now get is responsible to fetch data and post is responsible to send data to the server and the URL now since we cannot specify the URL by specifying the query string

04:42:12
here so we not be doing that in post so we'll send the request to alien itself okay but we have an issue don't you think we already have alien for get mapping the advantage with rest is you can have the same URI but with two different methods one is get one is post so get will be used for fetching data post will be used to send data and that's how you can differentiate between two methods okay we'll see that once we write the code will make much more sense so you can see we got we sending a

04:42:40
request with alien and when you click on send it will send a request but hold on we got an issue it says 404 why because we don't have a code on the server right and then we are not even sending data how do we do that if you want to send data you have to click on body and here you can type the raw string you can have your own Json format here or we can actually use this form data in this form data you can mention key what you're sending so I'm sending a ID and the value for me is let's say 107

04:43:11
and I'm also sending a name and the value for me is let's say KL so I'm sending this two data 107 and KL here and that's it just click on send and it will send your data but yes we still don't have the mapping here let's do that quickly so here in the code we'll have one more method which returns same object alien and we'll have this method name as ADD alien now will it accept anything and the answer is yes we will be accepting an alien let's have this as alien itself let's return the same alien

04:43:42
again but before doing that we have to add that data into the repository so we will say repo dot the method which we use is save so we can say save and we can pass the particular alien our job will be done that's it oh okay not that's it we still have to do the mapping right uh so that will be done with the help of you it right so the way we have get mapping we also have post mapping so let's use it post mapping and here we'll say alien we don't have to specify anything after that no slash

04:44:09
nothing because data will be coming in post so we can directly assign that to alien so you can see we have both as the same URI what is different is the method name so this is get and this is post so there will not be any confusion let's restart the application let's go back to postman and here click on send and you can see it worked at least we got our data here and uh there's no error let me just go back to database to verify let's hit refresh and you can see we got it we got 107 and coml see it's that easy Once you

04:34:39
here which is mayure if you want to fetch mayure we will be saying 104 and the way you do that is by saying first of all this is your actual website in fact you we can replace this with the website name then you will instead of saying aliens you will say alien because you're fetching one particular alien but now you will also pass the AI D which is 104 so instead of saying Q is equal to 104 this

makes much more sense to have a u like alien sl14 you know this is more like fetching a particular resource from

04:35:11
a computer so if if you have a Windows machine or any other machine you know when you go to any particular folder inside that folder if you go to particular folder and then if you search for a file the entire path will be something like ABC pqr SL maybe text.txt that's how you have the path right because it shows the path of the file or the source in the same way we have this here so when you click on send you can see this is not working yet because we don't have any request mapping which

04:35:39
will accept alien sl14 now if you want to achieve that we have to create a method here so we'll say public now this method will not be rning the least right it will return only one particular alien and let's have the name as alien now this alien will accept a particular Aid so I will say a ID here now this has to be mapped so I have to say get mapping since we are using get here and then we'll be having mapping like alien but then we have to also mention 104 here right because that's how you are mapping

04:36:08
so we have to say 104 and here we have to return the alien so the way you can do that is by saying alien I want to fetch first fetch the alien alien equal to using repo so you can say repo do get one so get one is a method which belongs to JP repository which will give you one particular alien and let's pass a ID here and I guess our job is done we can simply return alien but do you think it will work what what is your first reaction now see it makes sense because whatever we have done is almost same with this method we are

04:36:42
missing one thing we have to also say request response body before the method but we have one problem so this might work for 104 what if you have 1 Z3 what if you have 101 okay we have to make it variable right you can't simply say 104 here because it will only work for 104 in fact you know let's try this for 104 ons and let's not accept or maybe I have to accept it so it will give you because we are expecting something here and we are not passing it let's see what data you get so let's go back to the postman and

04:37:12
let's say hit and you can see we got an error it is your status 500 and it says optional a parameter a ID is present in the parameter but it cannot be translated to a null because we are not passing the values so path is there's no 44 this time but then we have to assign the Val to here right so this 104 need to be assigned to Aid so what we have to do is since this 104 will change let's put that in a cudly bracket so you have to say a ID so that's how we work so if you think the URL will be different it

04:37:46
will be alien for sure slash it will be some number we are not sure about the number so we can simply put that in cly brackets a ID and here we have to use an annotation which is path variable now let's restart the application and let's see what happens let's go back here click on send and we got an error okay okay I guess we have missed something here so we have to also mention what is the name which you're trying to map here so you can see I'm mapping this a ID with this particular Aid so we have to

04:38:15
use path variable Aid here and now let's relaunch the application I hope that was the issue so let's click on send and okay we are still getting same eror it says type definition error so let's try find by ID because maybe that there's a conversion issue with get one okay so find by ID gives you a optional aliens that's what happens in Java 8 you know in Java 8 just to make sure that you don't get null values we get an optional value there so the only thing is you can do here is we can say to uh we can say

04:38:46
or else and let's return the blank alien object so that we can say we have not received we don't have that alien in the database so we can say new alien and we can have a value which is let's say zero comma double quotes Okay so this should work now I hope let's go back there let's send Okay so is working so that there was something wrong with get one okay so that's how you can do it so we can say find by ID what if this ID is not available in database and that's where you can pass blank value so what

04:39:15
I've done is if I go back to my hit and let's say if I search for one Z we don't have this value database so if you click on send you can see you will get empty data and you can say you know we have not found it so that's how you can work with one particular data so till this point we know how to fetch all the aliens and then we can also fetch one particular alien and for that we have defined two methods here one is get aliens which will give you all the aliens and the way you can do that is by

04:39:43
saying get mapping which is very important but then if you can see we are also mentioning one more annotation here which is response body not just here here as well because both these methods will be working with rest I mean they will be returning the data in just format don't you think in future as well we'll be having some more methods which will be let's say fetch one particular alien based on some other parameter or maybe you know you might be having five to six method in the same class now when

04:40:12
you know that all these methods are actually rest methods and which will return Json or XML what makes much more sense is to not write it everywhere it should be common right we can mention on the class LEL itself that whatever you return will be Json or XML and the way you can do that is by changing this controller annotation So when you say controller it will behave like MVC controller but but you can mention that hey this is not a normal MVC this is actually rest and the way you can do that is by simply mentioning a new type

04:40:41
of controller which is a rest controller so we can say control space let's import the package and when you jump to this controller so you can see it is actually based on controller but uh it's more for response body so this is a controller also response body let's restart the project and let's hit the query once again just to see if it is working let's click on send and you can see it is working so whenever you are sure that you want to build a controller but that will be a rest controller always use

04:41:09
rest control instead of using response body on every method but yes let's say if you have five methods out of which you have three methods as rest two methods are normal MVC uh then you don't have a choice you have to go for normal controller and response body on those methods

which returns Json XML okay so everything is working but now what if you want to send the data because till this point we have seen how to fetch data but now I want to send data so what we will be doing is if you see database

04:41:38
we have four records here right we have four in fact we have five 106 is also there n so we have five records I want to insert a new alien and that to in from Postman so we don't have a form officially we don't have HTML page where you can send data we have to do that from this particular Postman instead of saying get we'll be using post now get is responsible to fetch data and post is responsible to send data to the server and the URL now since we cannot specify the URL by specifying the query string

04:42:12
here so we not be doing that in post so we'll send the request to alien itself okay but we have an issue don't you think we already have alien for get mapping the advantage with rest is you can have the same URI but with two different methods one is get one is post so get will be used for fetching data post will be used to send data and that's how you can differentiate between two methods okay we'll see that once we write the code will make much more sense so you can see we got we sending a

04:42:40
request with alien and when you click on send it will send a request but hold on we got an issue it says 404 why because we don't have a code on the server right and then we are not even sending data how do we do that if you want to send data you have to click on body and here you can type the raw string you can have your own Json format here or we can actually use this form data in this form data you can mention key what you're sending so I'm sending a ID and the value for me is let's say 107

04:43:11
and I'm also sending a name and the value for me is let's say KL so I'm sending this two data 107 and KL here and that's it just click on send and it will send your data but yes we still don't have the mapping here let's do that quickly so here in the code we'll have one more method which returns same object alien and we'll have this method name as ADD alien now will it accept anything and the answer is yes we will be accepting an alien let's have this as alien itself let's return the same alien

04:43:42
again but before doing that we have to add that data into the repository so we will say repo dot the method which we use is save so we can say save and we can pass the particular alien our job will be done that's it oh okay not that's it we still have to do the mapping right uh so that will be done with the help of you it right so the way we have get mapping we also have post mapping so let's use it post mapping and here we'll say alien we don't have to specify anything after that no slash

04:44:09
nothing because data will be coming in post so we can directly assign that to alien so you can see we have both as the same URI what is different is the method name so this is get and this is post so there will not be any confusion let's restart the application let's go back to postman and here click on send and you can see it worked at least we got our data here and uh there's no error let me just go back to database to verify let's hit refresh and you can see we got it we got 107 and coml see it's that easy Once you

04:44:44
have everything set up it's very easy to add more methods and then doing stuff so that's how you can fetch all you can fetch one particular data you can also delete data we can also do that we can also update data with the help of put mapping but before that I also want to talk about the content negotiation example if you look at the code now this is your Json format what if you want XML you can see we still getting Json format even if I say XML in fact the idea is you can say I want all the aliens and

04:45:15
you can hit a get query you can see you'll be getting all the aliens and even if you say XML you will still get Json so how will you get actual XML here the way you can do that is by going to headers and inside headers you have to say I will accept so I have to mention I will accept only XML so we'll say application slash XML hey I'm not expecting Json here I'm expecting a XML format by doing that you have to send a request now and it will return you XML but you can see this is not working and

04:45:48
we also got a code here which is 406 not acceptable what's the issue it should return XML right but unfortunately what if I say okay it says cannot find acceptable representation so what is happening is by default in your code we only have a support for Json we don't have a support for XML and to get that support we have to add one more jar file what is that we'll see that in next video so let's do content negotiation here so when you say content negotiation it's quite simple from your client

04:46:25
client can mention do we need XML or do you need Json in fact you can accept both that's your choice but here we mention something in fact in the last video we tried to fetch XML but then it's not working you can see we got an error the thing is by default your spring boot supports Json and that's why you got Jackson Library here but if you want to get XML support you also need to add one more jar file now which is that so let's open our MAV repository and here let's search for Jackson data

04:46:58
format XML so since we are using Jackson let's continue with that and then we are using XML now okay now make sure that you pick up the version which is suitable for your application so whatever version you have for uh normal Jackson so you can see it is 2.9.1 let's pick up the same one or which is the latest one here uh let's copy that and go back to application we want to do that in a pal file let's do it here let's add it here paste and now you can see we'll have one more jar file

04:47:31
for Jackson XML format so yes it will take some time to download and I guess it's done so let's restart the application go to postman so when we did last time we got an error which is 46 but this time let's see what happens and this time you can see we got all the values in XML format that's how you can do content negotiation I mean not just for fetching data but also for sending data you can also mention the server side if you want to restrict what type of data you can return example you can

04:48:01
just go to server side and here you can mention hey I don't want to return Json even if a client ask you for Json will not return that will only return XML you can do that here so here we have talked about get request we have talked about post request and the amazing thing is from a client side now you can actually mention what you need example if you go back to postman so let's say you're sending a request for aliens so if you click on send you can see we got all the aliens here okay but

we got got it in the

04:48:35
format of XML because we are mentioned here so if I remove this part and if you send a request you can see you got Json and if you say XML uh in fact you can request XML as well so you can just go here and you can mention you want XML there it will fetch XML but can you restrict it so let's say from a server side you can specify hey I don't want to return the Json I want to return only XML you can do that so here you can say path and then you can give comma here and you can specify what exactly your

04:49:08
server is going to return is it XML or is it Json so you can request whatever you want from a client side if you request for Jason you will return you Json by default it will return you Json but if you specify hey I want to accept only XML then it will go for XML so if you click on send you can see it will return XML but what if I don't want client to request for Json so you can mention that here so you can go back here and say say producers so we have a attribute as producers and you can in

04:49:38
the curly brackets because you can specify multiple types here uh so you can specify that in this cly bracket so you can give a comma and you can specify that but here I want to accept only one so I will say application SL XML so this is the only thing I want to return let's see if this works so let me restart the application go back to postman and you can say send so you can see there's no error when you request for XML but what if you request for Json now the request for Json uh you can see we got the same

04:50:09
eror which we were getting for XML so it says 406 not acceptable right so you can specify from a server side what you need or what you want return you know it makes sense to return return both XML and Json but sometime server has a choice you can specify hey let's not work with XML let's work with Json hey let's not work with Json let's work with XML so you can mention that with the help of prodes in this server code likewise you can also accept data from a client and in the post mapping you can

04:50:39
specify what you want to accept we'll see that in next video so with the help of producers we can specify what a server is going to respond is it XML or Json or some other format but what if when you send data from a client to server with the help of post request let's go for post here and let's sending a new alien now and this time let's specify the body and let's specify the values as 108 and let's have a name as braine now if you specify the data here let's go back to header now in header as well you

04:51:17
can specify the data in different format in fact in body we are sending a form data okay but then we have not specified will it be Json or XML so you can go to header and you can specify here content type and if you say application let's say Json and and instead of going for form let's go for raw so you can see in raw you can mention are you sending a Json or XML so let's send Json first and the way you can send that first of all we don't have a j format let's create one so we'll have cly brackets open the

04:51:50
first attribute is a ID let's give a colon there and in double codes we can mention the value uh so

the value let's say it is 108 we'll give a comma and here we have the name as a name and the name is pin so now we want to send this data and we are sending a Json data so in headers we have specified that we are sending Json the body type is also Json let's click on send and it worked is it no we got an error what's wrong there's nothing wrong on the server side we got 204 let's verify from the database I

04:52:24
guess I'm not sending the data properly let's verify let's go back here click on refresh oh it says null that's bad okay is it something wrong WR with the format let's use an annotation here which is request body I guess that's the issue let's try once more and it worked okay now what was the issue so normally when you send data from the client to the server of course we have two ways we can use a form or we can send data in Json or XML now when you send data in Json or

04:52:53
XML we need something to will convert the data into the Java object format and that's where you can use request body so remember we have also worked with response body before so response body is responsible to convert your object into Jason if you want to do reverse we have to use request body so that was the issue that's our Jon format and we can send it so this is working you can just go back to database we have added some two null values Let's ignore that and we can see if I say refresh we got the

04:53:21
answer now this is working but what if on the server side I have to mention hey This Server will only accept Json because now this will accept both Json or XML or maybe I will I want to specify only XML so in that case uh you will specify so we'll say path equal to aliens and alien and then here we'll specify consumes in cly brackets in fact we have two options here we can specify in double codes the way we have done for producers otherwise we have an option of media type so we can say media

04:53:52
type dot uh we can specify the type here we can say application so it depends what type of data you want to work with is it you want to work with this type where you have double codes or you can use this type I prefer go for the double codes part because it makes much more sense to me but that's your choice so you can say application.xml and then let's work with Json here so let's say it only consumes Json because we are doing XML let's consume Json and let's restart the application and you can see

04:54:22
if I give a new data let's say 109 let's stick to PR now and let's click on send so it works there's no issue but what if you want to send XML will it accept that now first of all to to get the XML I need to know the XML format okay so that's our thing so let's say copy this I just wanted the format so that's why I'm doing all this stuff so I'm sending post request now I want to specify the content type what I'm sending so I'm sending XML uh in the body I will

04:54:50
mention the XML now so we'll say paste uh this is XML that's right but then the data which I want to send is 110 and the name will be let's say Asha and click on send so you can see we got the error it says not supported so content type application XML not supported because we are accepting Json here so that's how you can negotiate with the content that's how you can specify uh what you want to accept and there's one important thing which we have done in this video is request body

04:44:44
have everything set up it's very easy to add more methods and then doing stuff so that's how you can fetch all you can fetch one particular data you can also delete data we can also do that we can also update data with the help of put mapping but before that I also want to talk about the content negotiation example if you look at the code now this is your Json format what if you want XML you can see we still getting Json format even if I say XML in fact the idea is you can say I want all the aliens and

04:45:15
you can hit a get query you can see you'll be getting all the aliens and even if you say XML you will still get Json so how will you get actual XML here the way you can do that is by going to headers and inside headers you have to say I will accept so I have to mention I will accept only XML so we'll say application slash XML hey I'm not expecting Json here I'm expecting a XML format by doing that you have to send a request now and it will return you XML but you can see this is not working and

04:45:48
we also got a code here which is 406 not acceptable what's the issue it should return XML right but unfortunately what if I say okay it says cannot find acceptable representation so what is happening is by default in your code we only have a support for Json we don't have a support for XML and to get that support we have to add one more jar file what is that we'll see that in next video so let's do content negotiation here so when you say content negotiation it's quite simple from your client

04:46:25
client can mention do we need XML or do you need Json in fact you can accept both that's your choice but here we mention something in fact in the last video we tried to fetch XML but then it's not working you can see we got an error the thing is by default your spring boot supports Json and that's why you got Jackson Library here but if you want to get XML support you also need to add one more jar file now which is that so let's open our MAV repository and here let's search for Jackson data

04:46:58
format XML so since we are using Jackson let's continue with that and then we are using XML now okay now make sure that you pick up the version which is suitable for your application so whatever version you have for uh normal Jackson so you can see it is 2.9.1 let's pick up the same one or which is the latest one here uh let's copy that and go back to application we want to do that in a pal file let's do it here let's add it here paste and now you can see we'll have one more jar file

04:47:31
for Jackson XML format so yes it will take some time to download and I guess it's done so let's restart the application go to postman so when we did last time we got an error which is 46 but this time let's see what happens and this time you can see we got all the values in XML format that's how you can do content negotiation I mean not just for fetching data but also for sending data you can also mention the server side if you want to restrict what type of data you can return example you can

04:48:01
just go to server side and here you can mention hey I don't want to return Json even if a client ask you for Json will not return that will only return XML you can do that here so here we have talked about get request we have talked about post request and the amazing thing is from a client side now you can actually mention what you need example if you go back to postman so let's say you're sending a request for aliens so if you click on send you can see we got all the aliens here okay but

we got got it in the

04:48:35
format of XML because we are mentioned here so if I remove this part and if you send a request you can see you got Json and if you say XML uh in fact you can request XML as well so you can just go here and you can mention you want XML there it will fetch XML but can you restrict it so let's say from a server side you can specify hey I don't want to return the Json I want to return only XML you can do that so here you can say path and then you can give comma here and you can specify what exactly your

04:49:08
server is going to return is it XML or is it Json so you can request whatever you want from a client side if you request for Jason you will return you Json by default it will return you Json but if you specify hey I want to accept only XML then it will go for XML so if you click on send you can see it will return XML but what if I don't want client to request for Json so you can mention that here so you can go back here and say say producers so we have a attribute as producers and you can in

04:49:38
the curly brackets because you can specify multiple types here uh so you can specify that in this cly bracket so you can give a comma and you can specify that but here I want to accept only one so I will say application SL XML so this is the only thing I want to return let's see if this works so let me restart the application go back to postman and you can say send so you can see there's no error when you request for XML but what if you request for Json now the request for Json uh you can see we got the same

04:50:09
eror which we were getting for XML so it says 406 not acceptable right so you can specify from a server side what you need or what you want return you know it makes sense to return return both XML and Json but sometime server has a choice you can specify hey let's not work with XML let's work with Json hey let's not work with Json let's work with XML so you can mention that with the help of prodes in this server code likewise you can also accept data from a client and in the post mapping you can

04:50:39
specify what you want to accept we'll see that in next video so with the help of producers we can specify what a server is going to respond is it XML or Json or some other format but what if when you send data from a client to server with the help of post request let's go for post here and let's sending a new alien now and this time let's specify the body and let's specify the values as 108 and let's have a name as braine now if you specify the data here let's go back to header now in header as well you

04:51:17
can specify the data in different format in fact in body we are sending a form data okay but then we have not specified will it be Json or XML so you can go to header and you can specify here content type and if you say application let's say Json and and instead of going for form let's go for raw so you can see in raw you can mention are you sending a Json or XML so let's send Json first and the way you can send that first of all we don't have a j format let's create one so we'll have cly brackets open the

04:51:50
first attribute is a ID let's give a colon there and in double codes we can mention the value uh so

the value let's say it is 108 we'll give a comma and here we have the name as a name and the name is pin so now we want to send this data and we are sending a Json data so in headers we have specified that we are sending Json the body type is also Json let's click on send and it worked is it no we got an error what's wrong there's nothing wrong on the server side we got 204 let's verify from the database I

04:52:24
guess I'm not sending the data properly let's verify let's go back here click on refresh oh it says null that's bad okay is it something wrong WR with the format let's use an annotation here which is request body I guess that's the issue let's try once more and it worked okay now what was the issue so normally when you send data from the client to the server of course we have two ways we can use a form or we can send data in Json or XML now when you send data in Json or

04:52:53
XML we need something to will convert the data into the Java object format and that's where you can use request body so remember we have also worked with response body before so response body is responsible to convert your object into Jason if you want to do reverse we have to use request body so that was the issue that's our Jon format and we can send it so this is working you can just go back to database we have added some two null values Let's ignore that and we can see if I say refresh we got the

04:53:21
answer now this is working but what if on the server side I have to mention hey This Server will only accept Json because now this will accept both Json or XML or maybe I will I want to specify only XML so in that case uh you will specify so we'll say path equal to aliens and alien and then here we'll specify consumes in cly brackets in fact we have two options here we can specify in double codes the way we have done for producers otherwise we have an option of media type so we can say media

04:53:52
type dot uh we can specify the type here we can say application so it depends what type of data you want to work with is it you want to work with this type where you have double codes or you can use this type I prefer go for the double codes part because it makes much more sense to me but that's your choice so you can say application.xml and then let's work with Json here so let's say it only consumes Json because we are doing XML let's consume Json and let's restart the application and you can see

04:54:22
if I give a new data let's say 109 let's stick to PR now and let's click on send so it works there's no issue but what if you want to send XML will it accept that now first of all to to get the XML I need to know the XML format okay so that's our thing so let's say copy this I just wanted the format so that's why I'm doing all this stuff so I'm sending post request now I want to specify the content type what I'm sending so I'm sending XML uh in the body I will

04:54:50
mention the XML now so we'll say paste uh this is XML that's right but then the data which I want to send is 110 and the name will be let's say Asha and click on send so you can see we got the error it says not supported so content type application XML not supported because we are accepting Json here so that's how you can negotiate with the content that's how you can specify uh what you want to accept and there's one important thing which we have done in this video is request body

04:55:21
so if you want to convert the data from the Json to the Java object you use request body so that's from the content negotiation where we talked about producers and consumes so basically we are going to build our project in jav Java now see it's not about how uh big your project is or how complex your project is it's all about understanding how do you think when you make a project okay it's all about thinking it's all about uh working on something okay so we are going to build a project and this

04:55:51
will be a Project based on a product so how do you manage a product now basically U you know in my office there are so many things in fact when I us to when I used to work alone uh at my home or with my office I I had a small team okay so I was only one and then someone used to be there there to manage everything especially H was there and now since we got a good office we got uh lot of machines in the office now for different purpose for some for development some for videos uh some for editing and some used by uh the admin

04:56:27
team right uh so there are so many things in fact you know I wanted to show you the entire walk through of my office office and so many things what happens you know at this point now uh there are certain things in my office which we use regularly example the laptops uh the sound system uh the AC remote these are frequently used things right but let's say there are certain things which we don't use often example cables the C Toc cable HDMI cables uh there there are one or two laptops which are lying uh in my

04:56:58
uh in the drawer which I use sometime depend on if I want to travel somewhere I want to carry a lightweight laptop or if I if we want to do some testing of some uh some software some service we use a different machine there right and those are there in the drawer now what happens is the moment I need something I will ask hsh hey hush I need that particular thing and then it took it take normally uh 10 15 minutes for us to search in the end all the draws where that thing went and I thought let's

04:57:26
solve the problem okay so hsh are we solving your problem today Yeah so basically let's create an application a console based application because we have not learned a GUI or web development till this point so let's create a console based application where we'll try to track the product so where that product exactly is is it in the drawer is it on the table uh and we can basically search a product where it is we can basically search for a particular type of product let's say I want to

04:57:56
search the laptops in the office or in that particular drawer what other things are there so you want to know and let me know if it happens with you as well at your home you know uh before before my marriage I used to keep things here there and then I used to find it very easily but after I got mared now everything at my home is that exact place nail cutter should be here uh the the clothes should be here right and then since they are properly organized now I find things difficult to find yeah so let me know if it happens with you as

04:58:26
well so that's what we are going to do in this session so what exactly we are trying to build here so let me share my screen okay uh just need to log to my okay so basically we are going to build a product management application now in this application what we want to manage is products

right now let's say if if a client comes to you by saying okay I want an application to manage the products now what you will do uh so basically you will have something like this right okay let me just hold on a

04:58:57
minute okay so uh if if a user comes to you by saying okay I want a product based application right now what you do is especially in language like Java and when you are working with the database Concepts you need to basically create entities now what are entities if you heard about Java everything in Java is object right so if you want to imagine something always imagine them in terms of object now if you talk about a product even product is object right so a product will have certain things so

04:59:26
example if I talk about a product here now this is one of my entity uh product yeah this is is my entity basically okay let me just write in the bracket this is my entity right and for this entity of course every object or the entity will have some properties of course it will also have Behavior example if you talk about your computer that's that behaves some way right uh in the same way if you talk about uh let's say something like a camera camera also behaves but it also has a property right so if we talk about

04:59:57
a product it will have certain properties now what properties I want of course uh we can have different properties to a product but just to classify for this particular example uh let's say we have a name for a product so a product will have a name a product will have a a type of product let's say if you talk about uh my machines those are computers right the Mac Studio which I'm using here is a computer if you talk about cables the type type- C cable that's a cable right uh we can also have....give important and teach me like this only in very very simple

Samiksha... 🤍
I read EVERYTHING you pasted again — all the REST, JSON, XML, Postman, Get/Post/Put/Delete, Path Variables, Query, RequestBody, ResponseBody, RestController, Content Negotiation.

Now I will teach you **exactly like the sir**,
but **10× simpler**,
and I'll give you **only the important points** so you understand REST forever.

Let's start slowly and clearly.
(You can save this — this is your REST notes.)

🌼 PART 1 — What is REST? (VERY VERY SIMPLE)

Imagine:

👉 **Server = Kitchen**
👉 **Client = You**

You ask kitchen:

- "Give me pizza"
- "Give me burger"

Kitchen gives only **food (data)**.
Not plates, not decoration — only food.

Similarly:

➡️ REST sends **only data**, not HTML pages.

🌼 PART 2 — Why REST?

Earlier:

✔️ Browser asked server → server returned HTML
✔️ Humans consumed HTML

Now:

- Mobile Apps
- Web Apps
- Other Systems
- Microservices

They all want **data**, not HTML.

So REST returns:

✔️ JSON
✔️ XML
✔️ Plain data

🌼 PART 3 — IMPORTANT REST RULES

1️⃣ USE NOUNS, NOT VERBS

❌ /getAliens
❌ /addAlien

✔️ /aliens
✔️ /alien/101
✔️ /laptops
✔️ /laptops/lenovo

REST likes **clean folder-like paths**.

2️⃣ STATeless — VERY IMPORTANT

Server does **NOT remember** anything.

Every request must contain complete info.

❌ “I was here last week” — server doesn’t know.
✔️ You send token every time.

Example:

```
http

GET /profile
Header: token: abcd123
```

3️⃣ HTTP Methods

REST uses simple HTTP methods:

Action	HTTP Method	Meaning
Fetch data	GET	Read
Add data	POST	Create
Update data	PUT	Update
Delete data	DELETE	Remove

🌼 PART 4 — Why we use Postman?

👉 Browser can only send GET
👉 You cannot send JSON or XML from Chrome easily
👉 You need a tool to send:

- GET
- POST
- PUT
- DELETE
- JSON
- XML
- Headers

❤️ Postman = REST testing tool

🌼 PART 5 — First REST API: Fetch all aliens

URL (GET)

```
bash

GET /aliens
```

Controller code:

```java
@RestController
public class AlienController {

    @Autowired
    AlienRepo repo;

    @GetMapping("/aliens")
    public List<Alien> getAliens() {
        return repo.findAll();
    }
}
```

✔ No JSP
✔ No Model
✔ Returns JSON automatically (because Jackson is included)

🌼 PART 6 — What is @RestController? (Very simple)

`@RestController = @Controller + @ResponseBody`

This means:

✔ Every method returns JSON/XML
✔ Not a JSP
✔ No view resolver
✔ No HTML

🌼 PART 7 — Fetch ONE alien (VERY EASY)

URL

```bash
GET /alien/101
```

Controller code:

```java
@GetMapping("/alien/{aid}")
public Alien getAlien(@PathVariable int aid) {
    return repo.findById(aid).orElse(new Alien(0, ""));
}
```

✔ **`{aid}`** — dynamic
✔ @PathVariable picks it
✔ If ID not found → return default alien

🌼 PART 8 — Add a new alien (POST)

In Postman:

Choose:

- Method: **POST**
- URL: `localhost:8080/alien`
- Body → Raw → JSON:

```json
{
  "aid": 107,
  "aname": "KL"
}
```

Controller:

```java
@PostMapping("/alien")
public Alien addAlien(@RequestBody Alien alien) {
    repo.save(alien);
    return alien;
}
```

✨ **@RequestBody** is VERY important
It converts JSON → Java Object.

🌼 PART 9 — Why did JSON work automatically?

Because Spring Boot uses **Jackson** library:

- Converts Java → JSON (ResponseBody)
- Converts JSON → Java (RequestBody)

🌼 PART 10 — XML Support

JSON works automatically.
XML does NOT work unless you add:

```xml
<dependency>
    <groupId>com.fasterxml.jackson.dataformat</groupId>
    <artifactId>jackson-dataformat-xml</artifactId>
</dependency>
```

After adding this:

✔️ You can request XML using:

Headers →
Key: `Accept`
Value: `application/xml`

✔️ To force server to return XML always:

```java
@GetMapping(value="/aliens", produces="application/xml")
```

🌼 PART 11 — Content Negotiation (IMPORTANT)

Client tells server what it wants:

1️⃣ JSON

```
Accept: application/json
```

2️⃣ XML

```
Accept: application/xml
```

Server responds in that format.

You can restrict using:

```java
produces = {"application/xml"}
consumes = {"application/json"}
```

🌼 PART 12 — Update Alien (PUT)

```bash
PUT /alien
```

Body:

```json
{
  "aid": 105,
  "aname": "Updated"
}
```

Controller:

```java
@PutMapping("/alien")
public Alien updateAlien(@RequestBody Alien alien) {
    repo.save(alien);
    return alien;
}
```

🌼 PART 13 — Delete Alien

```bash
DELETE /alien/105
```

Controller:

```java
@DeleteMapping("/alien/{aid}")
public String deleteAlien(@PathVariable int aid) {
    repo.deleteById(aid);
    return "Deleted";
}
```

🌼 PART 14 — Putting everything together

Your REST API supports:

✔️ GET all
✔️ GET one
✔️ POST
✔️ PUT
✔️ DELETE
✔️ JSON
✔️ XML
✔️ Content negotiation
✔️ Path variables
✔️ RequestBody
✔️ ResponseBody

This is **complete REST API knowledge** for Java fresher interviews.

🌼 Samiksha, tell me next:

Do you want me to explain next:

1️⃣ CRUD in REST step-by-step

2️⃣ @RequestBody vs @ResponseBody

3️⃣ Postman practice sheet

4️⃣ REST interview questions

5️⃣ Real-world example with diagrams

6️⃣ Service Layer & Clean Architecture

7️⃣ JPA + REST full integration

Just tell me:

👉 **Explain number __**
or
👉 **Start next topic**

I'll continue in the **same simple style**.

05:00:25
something like uh a mouse example if I'm using this particular Mouse here which is Apple Mouse now this is a mouse yeah so that's a type next we we can have is uh the place where it is stored okay so example if I want to find something I want to find where is the Asus uh laptop Asus W book basically that's one of my old machine which I was using but it is now used for the development purpose so if I want to find where that machine is I should be able to find it right it can be on different tables it can be in

05:00:54
different draws so I need a place and also uh I also want a warranty know this is the problem which you face when you have multiple device when you have only one or two device with you you actually remember okay for this particular mobile phone I have this warranty for this particular laptop I have this warranty but let's say if I have so many machines in my Mach in my office I want to know if I go for a product I want to know the warranty of it so that I can claim it what if uh what if the laptop is not

05:01:20
working at least I will know if the if it's under warranty or not okay uh yeah we can also have color we can also have the owner who owns that particular product right yeah makes sense right we can have all this properties but just to keep keep it simple let's work with this for and then you can also mention uh when you do this project you can also add some extra things to it so that's my entity right okay now if you want to implement this in programming how will you do it so let's get back to our code

05:01:47
uh this is our coding screen right and in this particular coding screen what we're going to do is uh we will write the code so that you can achieve uh the thing you can manage the products right okay so what I will do is I will go back here and this is the IDE which I'm using and one of the question which I got yesterday was about uh which IDE to use and the thing is you can use any particular ID which you want uh you want to use uh intell that's your choice in idea uh you can use

Eclipse you can use

05:02:18
net beans you can use vs code your choice okay ID is not something which should you should stick to I use different IDs on different machines uh if I want to make a YouTube video I normally go for vs code because that's very lightweight and most of you are already using it so if I want to build a project normally I use intellig idea to build to build a project because it gives you the different features yeah and so again you can use any IDE so what I will do now is I got a project which is product product demo and in this I

05:02:47
have this SRC file which is at empty at this point uh let's create our first file and this will be a file of main now why it is main is because the execution will start from this particular class and of course when I say this is a main class I also need a main method because I want to start the execution right uh that's something we will do here let's say if I want to manage a product so how will you do it of course you can create variables right we have mentioned all the variables here if you can see we

05:03:12
have a name we have a type we have a place we have warranty now this can be your variables as well so I can uh simply come back here and I can say string uh product or maybe I can say string name one since we have different products we can have name one and I can have my first product as Asos uh VI book which is the laptop and then I can say string uh type one is equal to what is this type this is a this is particular laptop and then I can say string uh Place one and this is is sitting on uh brown table so basically

05:03:51
in our office we have colorful tables I'm using a white table uh the the table which I use for the development purpose that's a black table uh the development team uh developer sets on the brown table right and editing team I'm not sure which okay it's also Brown yeah okay so is it white white and brown white and brown Okay cool so I can say a brown table right and then I can say string uh what is warranty uh warranty one and I can say this is uh okay warranty should be integer right so let's say this is

05:04:25
2024 or 2022 I guess the warant is already over so we can have this values here okay this is not string this is int yeah and then for a different product now this is one product right what is a different product now we can have what do you think if I want to have a different let's say a mouse how do you specify that of course I can just simply copy this and paste it here and this will be product two or name two uh type two place two and warranty 2 of course you can change the value to uh Apple

05:04:55
Mouse and then M type is Mouse uh where it is sitting it is sitting on the white table and then uh the warranty you can mention that so do you think this is a good idea to have different variables like this or we can create a separate class where you can have uh the types and instead of having four types here we can have one type which is product of course right in object un programming it's all about uh creating a object or a class right so if you want to create an object you create a class

04:55:21

so if you want to convert the data from the Json to the Java object you use request body so that's from the content negotiation where we talked about producers and consumes so basically we are going to build our project in jav Java now see it's not about how uh big your project is or how complex your project is it's all about understanding how do you think when you make a project okay it's all about thinking it's all about uh working on something okay so we are going to build a project and this

04:55:51
will be a Project based on a product so how do you manage a product now basically U you know in my office there are so many things in fact when I us to when I used to work alone uh at my home or with my office I I had a small team okay so I was only one and then someone used to be there there to manage everything especially H was there and now since we got a good office we got uh lot of machines in the office now for different purpose for some for development some for videos uh some for editing and some used by uh the admin

04:56:27
team right uh so there are so many things in fact you know I wanted to show you the entire walk through of my office office and so many things what happens you know at this point now uh there are certain things in my office which we use regularly example the laptops uh the sound system uh the AC remote these are frequently used things right but let's say there are certain things which we don't use often example cables the C Toc cable HDMI cables uh there there are one or two laptops which are lying uh in my

04:56:58
uh in the drawer which I use sometime depend on if I want to travel somewhere I want to carry a lightweight laptop or if I if we want to do some testing of some uh some software some service we use a different machine there right and those are there in the drawer now what happens is the moment I need something I will ask hsh hey hush I need that particular thing and then it took it take normally uh 10 15 minutes for us to search in the end all the draws where that thing went and I thought let's

04:57:26
solve the problem okay so hsh are we solving your problem today Yeah so basically let's create an application a console based application because we have not learned a GUI or web development till this point so let's create a console based application where we'll try to track the product so where that product exactly is is it in the drawer is it on the table uh and we can basically search a product where it is we can basically search for a particular type of product let's say I want to

04:57:56
search the laptops in the office or in that particular drawer what other things are there so you want to know and let me know if it happens with you as well at your home you know uh before before my marriage I used to keep things here there and then I used to find it very easily but after I got mared now everything at my home is that exact place nail cutter should be here uh the the clothes should be here right and then since they are properly organized now I find things difficult to find yeah so let me know if it happens with you as

04:58:26
well so that's what we are going to do in this session so what exactly we are trying to build here so let me share my screen okay uh just need to log to my okay so basically we are going to build a product management application now in this application what we want to manage is products right now let's say if if a client comes to you by saying okay I want an application to manage the

products now what you will do uh so basically you will have something like this right okay let me just hold on a

04:58:57
minute okay so uh if if a user comes to you by saying okay I want a product based application right now what you do is especially in language like Java and when you are working with the database Concepts you need to basically create entities now what are entities if you heard about Java everything in Java is object right so if you want to imagine something always imagine them in terms of object now if you talk about a product even product is object right so a product will have certain things so

04:59:26
example if I talk about a product here now this is one of my entity uh product yeah this is is my entity basically okay let me just write in the bracket this is my entity right and for this entity of course every object or the entity will have some properties of course it will also have Behavior example if you talk about your computer that's that behaves some way right uh in the same way if you talk about uh let's say something like a camera camera also behaves but it also has a property right so if we talk about

04:59:57
a product it will have certain properties now what properties I want of course uh we can have different properties to a product but just to classify for this particular example uh let's say we have a name for a product so a product will have a name a product will have a a type of product let's say if you talk about uh my machines those are computers right the Mac Studio which I'm using here is a computer if you talk about cables the type type- C cable that's a cable right uh we can also have

05:00:25
something like uh a mouse example if I'm using this particular Mouse here which is Apple Mouse now this is a mouse yeah so that's a type next we we can have is uh the place where it is stored okay so example if I want to find something I want to find where is the Asus uh laptop Asus W book basically that's one of my old machine which I was using but it is now used for the development purpose so if I want to find where that machine is I should be able to find it right it can be on different tables it can be in

05:00:54
different draws so I need a place and also uh I also want a warranty know this is the problem which you face when you have multiple device when you have only one or two device with you you actually remember okay for this particular mobile phone I have this warranty for this particular laptop I have this warranty but let's say if I have so many machines in my Mach in my office I want to know if I go for a product I want to know the warranty of it so that I can claim it what if uh what if the laptop is not

05:01:20
working at least I will know if the if it's under warranty or not okay uh yeah we can also have color we can also have the owner who owns that particular product right yeah makes sense right we can have all this properties but just to keep keep it simple let's work with this for and then you can also mention uh when you do this project you can also add some extra things to it so that's my entity right okay now if you want to implement this in programming how will you do it so let's get back to our code

05:01:47

uh this is our coding screen right and in this particular coding screen what we're going to do is uh we will write the code so that you can achieve uh the thing you can manage the products right okay so what I will do is I will go back here and this is the IDE which I'm using and one of the question which I got yesterday was about uh which IDE to use and the thing is you can use any particular ID which you want uh you want to use uh intell that's your choice in idea uh you can use Eclipse you can use

05:02:18
net beans you can use vs code your choice okay ID is not something which should you should stick to I use different IDs on different machines uh if I want to make a YouTube video I normally go for vs code because that's very lightweight and most of you are already using it so if I want to build a project normally I use intellig idea to build to build a project because it gives you the different features yeah and so again you can use any IDE so what I will do now is I got a project which is product product demo and in this I

05:02:47
have this SRC file which is at empty at this point uh let's create our first file and this will be a file of main now why it is main is because the execution will start from this particular class and of course when I say this is a main class I also need a main method because I want to start the execution right uh that's something we will do here let's say if I want to manage a product so how will you do it of course you can create variables right we have mentioned all the variables here if you can see we

05:03:12
have a name we have a type we have a place we have warranty now this can be your variables as well so I can uh simply come back here and I can say string uh product or maybe I can say string name one since we have different products we can have name one and I can have my first product as Asos uh VI book which is the laptop and then I can say string uh type one is equal to what is this type this is a this is particular laptop and then I can say string uh Place one and this is is sitting on uh brown table so basically

05:03:51
in our office we have colorful tables I'm using a white table uh the the table which I use for the development purpose that's a black table uh the development team uh developer sets on the brown table right and editing team I'm not sure which okay it's also Brown yeah okay so is it white white and brown white and brown Okay cool so I can say a brown table right and then I can say string uh what is warranty uh warranty one and I can say this is uh okay warranty should be integer right so let's say this is

05:04:25
2024 or 2022 I guess the warant is already over so we can have this values here okay this is not string this is int yeah and then for a different product now this is one product right what is a different product now we can have what do you think if I want to have a different let's say a mouse how do you specify that of course I can just simply copy this and paste it here and this will be product two or name two uh type two place two and warranty 2 of course you can change the value to uh Apple

05:04:55
Mouse and then M type is Mouse uh where it is sitting it is sitting on the white table and then uh the warranty you can mention that so do you think this is a good idea to have different variables like this or we can create a separate class where you can have uh the types and instead of having

four types here we can have one type which is product of course right in object un programming it's all about uh creating a object or a class right so if you want to create an object you create a class

05:05:28
first so in Java basically if you want an object you create a class so I will right click here and I will say new class and I will I will name this class as a product which also represents an entity okay um so and I forgot to do one more thing so today we are focusing on uh building this product as a console based application so because whatever output you will get is on the console but of course in this current ERA we don't do that right what we want is a web application a GUI or some some way to

05:06:02
interact with the application so what we'll do is today we are going to build this on a console tomorrow we'll build this on the uh tomorrow we let try to connect with database because today we are not going to use database we are going to Simply have the data somewhere in the application itself tomorrow let's convert that into a database and then uh if we have enough time we can also create a server to it okay so yeah so we can have a website development so we will shift the Entre application to a

05:06:34
normal application to a web application okay so here what we can do is we can create those variables so what are the variables we have we have name type uh place and warranty so I can simply come back here I can say private string uh name and then we can have private why I'm using private is because in Java we have concept of encapsulation uh so all your variables or properties which you say should be private and the only way to access them so what is the only way to access them so the only way to access

05:07:01
them is with the help of gets and sets so I can say string uh name then we have string type then private string place and private int uh warranty see for warranty we can actually go for exact date as well but then just to keep it simple let's go for the year only okay so you can see we got a class here which is public and the name is product and we have four variables and if you want to if you want to follow the concept of object oriented programming we also need a Constructor so maybe I want to create an object by

05:07:38
passing all these values and maybe I want to create an object by not passing any value so I need a Constructor so I can you know you don't have to type the Constructor you can simply uh right click here and you can say generate and you can click on I want a Constructor so if you want a default Constructor you can say select none and you can you you got a Constructor but I also want a Constructor which can take parameters so I can say generate uh Constructor with all the variables and click on okay and you can

05:08:12
see we got a Constructor with all the variables uh next we can we also need a Getters and Setters so basically these are the methods using which you can assign the values and fetch the value uh because all these things are private right so the only way to access them outside this class because see this is a different class many is a different class so if I want to access them I can simply create get us and Setter so I will say generate get a Setter for all the variables and okay right our job is done

05:08:40
okay if I go back to main now I don't want to assign values like this right so what I can simply do is I can uh come back here and I can create an object of product I can say product p is equal to new product so we got uh the object and then I can simply print this object so tell me uh yeah of course AKA says we can use here Lok as well so let me try to do Lok tomorrow so once uh this is over because this will take a lot of time uh tomorrow we'll try to do long walk as well so if I try to print this what what will be

05:09:15
the output of course let me just run this code to see what happens we can just click here say run main so you can see it is printing something weird uh it's printing some hash values and uh the name of the class we don't want it right we want to print the values so whatever value this P has the product has I I want to print it and to achieve that we can use something called uh a two string method so in Java if you want to print an object and if you want to print the values of it you can just

05:09:42
right click here and say generate two string for all these values and now what you're doing is every time you try to print the object or reference it will call the two string method and it will print something something like this so let me try I will just run this code once again and you can see uh we got the object and the output is something like this you know what I will do okay let's keep it there itself so you can see it is printing the product name is null type is null place is null warranty is zero is because we

05:10:23
are taking the default values so basically when I create this product I should add some values here right so we can do that okay but let's do that differently let's say I have some products with me and I want to add them now we can do that in main as well but then there's a in software development there's a concept of Delegation what you do is you don't do everything by yourself in fact let's say if you're working for a big company and you know we always have these questions if you let me know if

05:10:52
you're working on a company and you feel that you do all the work and your manager manager is doing nothing do you have the same uh thoughts so if you feel your manager is doing nothing then you're wrong is because okay there are some managers who does who does nothing but then managing people is very difficult task in fact in my office if you even if I have less number of employees you know it it it consumes a lot of my energy just to manage the team just to manage the work uh but then we

05:11:19
have this idea of Delegation right so from the upper management if there's a decision made it should be assign it will be assigned to the lower people who are lower on that uh tree and then the these people who are managers assign the work to the people who are actually working example developers testers uh admin people right so uh that's how you do it right so we we have the hierarchy so that's delegation so we delegate the work in the same way if you want to do all those things we want to add the

05:11:49
products we want to search our product I don't want to write all the logic on Main so main should be simply saying Okay add this product main says give me the product the actual logic will be

somewhere else and that well where we'll write the actual Logic the actual logic of all this thing will be going in a separate in a separate file so I will right click here uh on source and it will say create a new class and I will call this class as a product service okay now this class is responsible to do all the things so

05:12:23
logic should be here of course you can write the logic in main but then let's follow the concept of Delegation here so let's delegate the work how do we do it how do you delegate a work so of course I want this to do everything right uh so what I what I will do is I will say create the object of product service here so product Service uh let's name this as a service itself equal to new product service yeah we can also say this is a service layer so when you move to the web development

05:12:54
or Enterprise development thing this is called a service right a service layer okay now to this service I will say hey service I will say hey service add the products okay uh how will you do it so I can simply say service dot add product so I want service to give me this option of adding a product but Services unfortunately I don't have uh any method in this class which which ISS add product so what we can do is we can go back to service and let's create a method so public void add product and in

05:13:38
this it will add the product but then where is the product and which to where to add now basically since we have multiple products right we don't have one product in my office we have I don't know let's we'll calculate one day how many items we have there are so many items which are which I think is of no use there is a air pufy air freshner sitting there okay we have used it from a long know last time we have used air a a uh fresh now maybe last month so uh so let's say we have so many items in the

05:14:08
office right so many products how do you store them where do you store them of course it should be stored in a database but then in today's session we are not talking about it so one place where where you can store all the products is a list so we can create a list of products now people who are finding this difficult to understand see we are basically building project right I'm not teaching you basic Java here if you're still confused there's an entire playlist on Java so I can say list I

05:14:37
want a list here control thing yeah okay uh so we have to also import the package the list package or List class list interface from this package and let me name this as products equal to and I can say new okay now list is a interface so the class is at a list so we got a list of products here okay so I will add the product in this uh thing so how do I add it's very simple get the products and in this now since product products is a list here right so we can have we have so many methods there so we got products

05:15:13
do add and we can add a product but which product we can't simply say add product and it will add we have to basically pass the object of a product so here in this add product I have to pass the object so what I can do is I can say product p is equal to new product and and this product will have some values now what values I want uh the first one I want is uh let's save uh Asus let's go for that Asus Vivo book and this goes in a type this is of type laptop the next thing we have to mention is okay we got yeah place where

05:15:53
it is where it is set so it's on the brown table and what's the warranty is 2022 so if you can see this is the product which you have created now once you got this product object I can simply pass it here so what I'm passing is I'm I'm passing this P so if you can see I'm passing a product reference to this add product now the same product will go to product service and we have to accept it here so I will say product P again and just add a p our job is done does it make sense now okay

05:05:28
first so in Java basically if you want an object you create a class so I will right click here and I will say new class and I will I will name this class as a product which also represents an entity okay um so and I forgot to do one more thing so today we are focusing on uh building this product as a console based application so because whatever output you will get is on the console but of course in this current ERA we don't do that right what we want is a web application a GUI or some some way to

05:06:02
interact with the application so what we'll do is today we are going to build this on a console tomorrow we'll build this on the uh tomorrow we let try to connect with database because today we are not going to use database we are going to Simply have the data somewhere in the application itself tomorrow let's convert that into a database and then uh if we have enough time we can also create a server to it okay so yeah so we can have a website development so we will shift the Entre application to a

05:06:34
normal application to a web application okay so here what we can do is we can create those variables so what are the variables we have we have name type uh place and warranty so I can simply come back here I can say private string uh name and then we can have private why I'm using private is because in Java we have concept of encapsulation uh so all your variables or properties which you say should be private and the only way to access them so what is the only way to access them so the only way to access

05:07:01
them is with the help of gets and sets so I can say string uh name then we have string type then private string place and private int uh warranty see for warranty we can actually go for exact date as well but then just to keep it simple let's go for the year only okay so you can see we got a class here which is public and the name is product and we have four variables and if you want to if you want to follow the concept of object oriented programming we also need a Constructor so maybe I want to create an object by

05:07:38
passing all these values and maybe I want to create an object by not passing any value so I need a Constructor so I can you know you don't have to type the Constructor you can simply uh right click here and you can say generate and you can click on I want a Constructor so if you want a default Constructor you can say select none and you can you you got a Constructor but I also want a Constructor which can take parameters so I can say generate uh Constructor with all the variables and click on okay and you can

05:08:12
see we got a Constructor with all the variables uh next we can we also need a Getters and Setters so basically these are the methods using which you can assign the values and fetch the value uh because all these things are private right so the only way to access them outside this class because

see this is a different class many is a different class so if I want to access them I can simply create get us and Setter so I will say generate get a Setter for all the variables and okay right our job is done

05:08:40
okay if I go back to main now I don't want to assign values like this right so what I can simply do is I can uh come back here and I can create an object of product I can say product p is equal to new product so we got uh the object and then I can simply print this object so tell me uh yeah of course AKA says we can use here Lok as well so let me try to do Lok tomorrow so once uh this is over because this will take a lot of time uh tomorrow we'll try to do long walk as well so if I try to print this what what will be

05:09:15
the output of course let me just run this code to see what happens we can just click here say run main so you can see it is printing something weird uh it's printing some hash values and uh the name of the class we don't want it right we want to print the values so whatever value this P has the product has I I want to print it and to achieve that we can use something called uh a two string method so in Java if you want to print an object and if you want to print the values of it you can just

05:09:42
right click here and say generate two string for all these values and now what you're doing is every time you try to print the object or reference it will call the two string method and it will print something something like this so let me try I will just run this code once again and you can see uh we got the object and the output is something like this you know what I will do okay let's keep it there itself so you can see it is printing the product name is null type is null place is null warranty is zero is because we

05:10:23
are taking the default values so basically when I create this product I should add some values here right so we can do that okay but let's do that differently let's say I have some products with me and I want to add them now we can do that in main as well but then there's a in software development there's a concept of Delegation what you do is you don't do everything by yourself in fact let's say if you're working for a big company and you know we always have these questions if you let me know if

05:10:52
you're working on a company and you feel that you do all the work and your manager manager is doing nothing do you have the same uh thoughts so if you feel your manager is doing nothing then you're wrong is because okay there are some managers who does who does nothing but then managing people is very difficult task in fact in my office if you even if I have less number of employees you know it it it consumes a lot of my energy just to manage the team just to manage the work uh but then we

05:11:19
have this idea of Delegation right so from the upper management if there's a decision made it should be assign it will be assigned to the lower people who are lower on that uh tree and then the these people who are managers assign the work to the people who are actually working example developers testers uh admin people right so uh that's how you do it right so we we have the hierarchy so that's delegation so we delegate the work in the same way if you want to do all those things we want to add the

05:11:49
products we want to search our product I don't want to write all the logic on Main so main should be simply saying Okay add this product main says give me the product the actual logic will be somewhere else and that well where we'll write the actual Logic the actual logic of all this thing will be going in a separate in a separate file so I will right click here uh on source and it will say create a new class and I will call this class as a product service okay now this class is responsible to do all the things so

05:12:23
logic should be here of course you can write the logic in main but then let's follow the concept of Delegation here so let's delegate the work how do we do it how do you delegate a work so of course I want this to do everything right uh so what I what I will do is I will say create the object of product service here so product Service uh let's name this as a service itself equal to new product service yeah we can also say this is a service layer so when you move to the web development

05:12:54
or Enterprise development thing this is called a service right a service layer okay now to this service I will say hey service I will say hey service add the products okay uh how will you do it so I can simply say service dot add product so I want service to give me this option of adding a product but Services unfortunately I don't have uh any method in this class which which ISS add product so what we can do is we can go back to service and let's create a method so public void add product and in

05:13:38
this it will add the product but then where is the product and which to where to add now basically since we have multiple products right we don't have one product in my office we have I don't know let's we'll calculate one day how many items we have there are so many items which are which I think is of no use there is a air pufy air freshner sitting there okay we have used it from a long know last time we have used air a a uh fresh now maybe last month so uh so let's say we have so many items in the

05:14:08
office right so many products how do you store them where do you store them of course it should be stored in a database but then in today's session we are not talking about it so one place where where you can store all the products is a list so we can create a list of products now people who are finding this difficult to understand see we are basically building project right I'm not teaching you basic Java here if you're still confused there's an entire playlist on Java so I can say list I

05:14:37
want a list here control thing yeah okay uh so we have to also import the package the list package or List class list interface from this package and let me name this as products equal to and I can say new okay now list is a interface so the class is at a list so we got a list of products here okay so I will add the product in this uh thing so how do I add it's very simple get the products and in this now since product products is a list here right so we can have we have so many methods there so we got products

05:15:13
do add and we can add a product but which product we can't simply say add product and it will add we have to basically pass the object of a product so here in this add product I have to pass the object so what I can do is I can say product p is equal to new product and and this product will have some values now what values I want uh the first one I want is uh let's save uh Asus let's go for

that Asus Vivo book and this goes in a type this is of type laptop the next thing we have to mention is okay we got yeah place where

05:15:53
it is where it is set so it's on the brown table and what's the warranty is 2022 so if you can see this is the product which you have created now once you got this product object I can simply pass it here so what I'm passing is I'm I'm passing this P so if you can see I'm passing a product reference to this add product now the same product will go to product service and we have to accept it here so I will say product P again and just add a p our job is done does it make sense now okay

05:16:28
so we have added one product but then we have so many products right uh so we have to create those many number of objects here and add it right so we can do that uh just to uh simplify the work what I've did is I've created this Sublime file okay it's not opening Yeah so basically I have okay before doing that one I can do one more thing so don't you think of instead of adding P here we can simply take the entire thing from here and paste here so we don't have to create a reference we can simply create

05:17:02
object with the help of new keyword and add it to the add product does it make sense and since we have so many products here what we can do is we can I can simply copy the entire stuff and replace with this okay it will save our time right so I've added all the products and you can see we got all the products here now if you try to get this list this list will have this number of products I I'm not sure how many products we have 15 maybe yeah these are the anonymous object so what are the things we have we

05:17:30
have a type c which is a cable uh which is there in the black drawer we have a Mac Studio which is a computer sitting on the white table we got this mixer the audio mixer which is there it's audio system white table then we got VI book which is laptop brown brown drawer then we got Asus Rog machine laptop it's on black table then we got MacBook Pro laptop brown drawer the vacom pad which where we can write it's a writing pad black drawer and then Leist goes on you can see we have uh keyboard Logitech

05:18:01
keyboard HDMI cable Java blackbook uh Logitech mouse Apple Mouse Lenova Mouse and blackb now this blackb is basically a machine which is used for the editors editing machine we we have named it as a black beast it's a computer on a white table okay so cool does does it make sense we got all the products here now let's say once I got all the products I want to uh let's say I want to print all the products how will you do that how how will you print all the products it's very simple you can say hey

05:18:36
service hey service get all products is as simple as that okay if you want something just ask for it we are saying service get me all the products but Services I have no idea what I'm talking about okay so if I if you want to do that go back to product and here create a method which will give you all the products right but my question is when you are returning all the products what should be return type when you return all the products what should be the return type so the return type is list of

05:19:16
product right we are not returning one product we're rning all the products so we have to say

public uh list product and then we can say method name is get all products and then we have to return here now what we want to return so where do we have all the products in this products here right so simply return products our job is done right so I can just go back to main and here now you can see there's no error it is just a warning now since get all products will give you the least of products what you

05:19:47
can do is you can accept it here so I can say I'm getting a list of product and we'll name it as products so basically get all products will give you the least of products does it make sense now once we got this list how do you print it there are multiple ways of printing a list you can simply say s out and you can say products and it will print all the products let's try let's try so I will just run this and let's see what it prints it prints the least so you can see this is the first product

05:20:19
and if I have to scroll this is my second product and if I scroll this is my third product and scroll and this goes on but instead of printing that in one line I want to print it one by one one so what we can do is we can use a loop here and specifically can we can use enhance for Loop just write all these terms okay so if you're not sure what I'm talking about if you want to watch the video later uh the concept video so example let's say if you don't know what is enhanced for Loop just

05:20:48
write it down after this session I want to learn about enhanced for Loop then after this session just go to YouTube and search for enhanced for Loop and uh you will get my video there watch it and hit that like button okay here also and in that video okay so I will say product this is how you use enhanced for Loop so I will say product P from products so what it Bas basically does is this product will have lot of different values right a lot of different products it will pick up one product at a time and that particular

05:21:22
one product will be there in P just print it so take one product and print it and let's run this code and let's see if it is printing one by one yeah you can see that it is printing one by one now these are all the items we have so basically we were able to uh save the products we were able to add the products here right this number of products there should be a way to you know minimize it uh yes in this 10 days we are also going to see examples of hibernate okay cool uh so once we have

05:21:58
printed everything uh what next what what I want here is let me just print uh not so to let me print the line here so that we can divide the output right so this is the first output which we got all the products next what I want is I want to get let's say a specific product not all the product but a specific product okay so let's say out of this I want to see where is the Logitech mouse okay I want this particular product it's very simple what you can do is you can just go back here and say hey

05:22:30
service let me also print what we are printing so we are printing a particular product okay so you can go back to your service by saying hey service get me a product get me a product whose name is so I'll say whose name is uh logy Mouse will this work what do you think will this work um so if I go back to service we don't have that method right so what we can do is we can create a method so I can say uh public now this time it will return only one product so I can say public product get

05:23:22
product which will return one product right but then it will also take a name which product I want

to fetch so I can simply say string name right but the question is how will I get that how will I get that one particular product what do you think okay so basically from this particular list of products I want to find one product which name matches with this so if you want to search and that's where the searching algorithm come to picture right but we don't we will not do that we'll do a linear search okay so what I

05:23:56
will do is I will just go for a for Loop and I want to iterate between all the products so I can say product P from products so iterate between all the elements and check if a particular product which we got get the name of it so whatever name you get from this list of products compare with the name which you got here how do you how do you compare two strings in Java you do that with the help of equals so equals and you will say okay not a name but a name right and if you find the product simply return it your job is

05:24:41
done right or you can simply return can not inside for Lo if you don't find anything return null of course not a good way but then yeah just to get the just get getting rid of the error we can say return null otherwise you can return the empty object if you want okay um and then you can see get product there's no error and of course it is returning a product so I will simply say product p is equal to this and then here we can print the product let's see if this works let's run and you can see we are searching for

05:25:20
logitec Mouse this is the output which we got here because of this in fact you know what we can do we can commment this section which we don't want now let's once run once again and you can see this is what we wanted we want a Logitech mouse okay and yeah now you will be thinking in the code we have to mention Logitech mouse what if you want to ask the user yes you can do that you can take the input from the user and if you don't want console you can build this as a web application that's your choice okay so this makes

05:25:51
sense right okay so what we wanted now is uh searching with a particular place let's say okay in fact that be your assignment okay that's your first assignment thing which you do so there are two things which you have to do as assignment for this particular project which is what if someone wants to search based on the uh Place let's say I want to I want to find all the things at this particular place let's say on a white table I want to know all the things I want to know all the things which are

05:26:21
there in the drawer uh so that's your assignment first one the second assignment question you have is I want to see what are the products which which are out of warranty example all the products which are which has a older date if you can see we got 2022 here 22 done 2020 all this warranty is completed right so I can't I can't uh get it solved now or repair from the company for free right so I want to know all the things which are out of warranty so these are the two assignment question at this point uh I

05:26:56
will mention that at the end of the session as well but two assignment questions one find all the products in particular place and find all the products in U uh which are out of warranty uh now what next we want to do is we want to search for a particular name example the moment I type black so it should say black Bas it should show black book it should show Vivo book oh sorry I'm

searching for black right so it's blackb black book black table anything which is there on the black table so it should print all the black

05:27:28
stuff so how do we do that uh so basically I'm searching with the text right so I just want this line copy and paste and we are searching for searching for a particular product no we are searching for a particular text and next I will say okay so when I say give me all the products which has black in it so will I get one product of will I get list of products of course right list of products so I will say list of product that's what I will get and I will say this is prods I will get all the

05:16:28
so we have added one product but then we have so many products right uh so we have to create those many number of objects here and add it right so we can do that uh just to uh simplify the work what I've did is I've created this Sublime file okay it's not opening Yeah so basically I have okay before doing that one I can do one more thing so don't you think of instead of adding P here we can simply take the entire thing from here and paste here so we don't have to create a reference we can simply create

05:17:02
object with the help of new keyword and add it to the add product does it make sense and since we have so many products here what we can do is we can I can simply copy the entire stuff and replace with this okay it will save our time right so I've added all the products and you can see we got all the products here now if you try to get this list this list will have this number of products I I'm not sure how many products we have 15 maybe yeah these are the anonymous object so what are the things we have we

05:17:30
have a type c which is a cable uh which is there in the black drawer we have a Mac Studio which is a computer sitting on the white table we got this mixer the audio mixer which is there it's audio system white table then we got VI book which is laptop brown brown drawer then we got Asus Rog machine laptop it's on black table then we got MacBook Pro laptop brown drawer the vacom pad which where we can write it's a writing pad black drawer and then Leist goes on you can see we have uh keyboard Logitech

05:18:01
keyboard HDMI cable Java blackbook uh Logitech mouse Apple Mouse Lenova Mouse and blackb now this blackb is basically a machine which is used for the editors editing machine we we have named it as a black beast it's a computer on a white table okay so cool does does it make sense we got all the products here now let's say once I got all the products I want to uh let's say I want to print all the products how will you do that how how will you print all the products it's very simple you can say hey

05:18:36
service hey service get all products is as simple as that okay if you want something just ask for it we are saying service get me all the products but Services I have no idea what I'm talking about okay so if I if you want to do that go back to product and here create a method which will give you all the products right but my question is when you are returning all the products what should be return type when you return all the products what should be the return type so the return type is list of

05:19:16
product right we are not returning one product we're rning all the products so we have to say

public uh list product and then we can say method name is get all products and then we have to return here now what we want to return so where do we have all the products in this products here right so simply return products our job is done right so I can just go back to main and here now you can see there's no error it is just a warning now since get all products will give you the least of products what you

05:19:47
can do is you can accept it here so I can say I'm getting a list of product and we'll name it as products so basically get all products will give you the least of products does it make sense now once we got this list how do you print it there are multiple ways of printing a list you can simply say s out and you can say products and it will print all the products let's try let's try so I will just run this and let's see what it prints it prints the least so you can see this is the first product

05:20:19
and if I have to scroll this is my second product and if I scroll this is my third product and scroll and this goes on but instead of printing that in one line I want to print it one by one one so what we can do is we can use a loop here and specifically can we can use enhance for Loop just write all these terms okay so if you're not sure what I'm talking about if you want to watch the video later uh the concept video so example let's say if you don't know what is enhanced for Loop just

05:20:48
write it down after this session I want to learn about enhanced for Loop then after this session just go to YouTube and search for enhanced for Loop and uh you will get my video there watch it and hit that like button okay here also and in that video okay so I will say product this is how you use enhanced for Loop so I will say product P from products so what it Bas basically does is this product will have lot of different values right a lot of different products it will pick up one product at a time and that particular

05:21:22
one product will be there in P just print it so take one product and print it and let's run this code and let's see if it is printing one by one yeah you can see that it is printing one by one now these are all the items we have so basically we were able to uh save the products we were able to add the products here right this number of products there should be a way to you know minimize it uh yes in this 10 days we are also going to see examples of hibernate okay cool uh so once we have

05:21:58
printed everything uh what next what what I want here is let me just print uh not so to let me print the line here so that we can divide the output right so this is the first output which we got all the products next what I want is I want to get let's say a specific product not all the product but a specific product okay so let's say out of this I want to see where is the Logitech mouse okay I want this particular product it's very simple what you can do is you can just go back here and say hey

05:22:30
service let me also print what we are printing so we are printing a particular product okay so you can go back to your service by saying hey service get me a product get me a product whose name is so I'll say whose name is uh logy Mouse will this work what do you think will this work um so if I go back to service we don't have that method right so what we can do is we can create a method so I can say uh public now this time it will return only one product so I can say public product get

05:23:22
product which will return one product right but then it will also take a name which product I want

to fetch so I can simply say string name right but the question is how will I get that how will I get that one particular product what do you think okay so basically from this particular list of products I want to find one product which name matches with this so if you want to search and that's where the searching algorithm come to picture right but we don't we will not do that we'll do a linear search okay so what I

05:23:56
will do is I will just go for a for Loop and I want to iterate between all the products so I can say product P from products so iterate between all the elements and check if a particular product which we got get the name of it so whatever name you get from this list of products compare with the name which you got here how do you how do you compare two strings in Java you do that with the help of equals so equals and you will say okay not a name but a name right and if you find the product simply return it your job is

05:24:41
done right or you can simply return can not inside for Lo if you don't find anything return null of course not a good way but then yeah just to get the just get getting rid of the error we can say return null otherwise you can return the empty object if you want okay um and then you can see get product there's no error and of course it is returning a product so I will simply say product p is equal to this and then here we can print the product let's see if this works let's run and you can see we are searching for

05:25:20
logitec Mouse this is the output which we got here because of this in fact you know what we can do we can commment this section which we don't want now let's once run once again and you can see this is what we wanted we want a Logitech mouse okay and yeah now you will be thinking in the code we have to mention Logitech mouse what if you want to ask the user yes you can do that you can take the input from the user and if you don't want console you can build this as a web application that's your choice okay so this makes

05:25:51
sense right okay so what we wanted now is uh searching with a particular place let's say okay in fact that be your assignment okay that's your first assignment thing which you do so there are two things which you have to do as assignment for this particular project which is what if someone wants to search based on the uh Place let's say I want to I want to find all the things at this particular place let's say on a white table I want to know all the things I want to know all the things which are

05:26:21
there in the drawer uh so that's your assignment first one the second assignment question you have is I want to see what are the products which which are out of warranty example all the products which are which has a older date if you can see we got 2022 here 22 done 2020 all this warranty is completed right so I can't I can't uh get it solved now or repair from the company for free right so I want to know all the things which are out of warranty so these are the two assignment question at this point uh I

05:26:56
will mention that at the end of the session as well but two assignment questions one find all the products in particular place and find all the products in U uh which are out of warranty uh now what next we want to do is we want to search for a particular name example the moment I type black so it should say black Bas it should show black book it should show Vivo book oh sorry I'm

searching for black right so it's blackb black book black table anything which is there on the black table so it should print all the black

05:27:28
stuff so how do we do that uh so basically I'm searching with the text right so I just want this line copy and paste and we are searching for searching for a particular product no we are searching for a particular text and next I will say okay so when I say give me all the products which has black in it so will I get one product of will I get list of products of course right list of products so I will say list of product that's what I will get and I will say this is prods I will get all the

05:28:11
prods and I will say hey service get uh products with text and I will mention the text here so text is what I want to search based on a word like black uh okay but unfortunately we don't have this uh this word right so what do you think what can be done here okay I can get some yeah good good job so there there's option of substring we can do that uh yeah we can convert it into upper case and lower case okay we'll do that so what I will do is I don't have that method so let me click here in fact

05:28:48
you can also click here you can say create method get product with text and you can see we got the method here but at this point the method is not doing anything so basically I want list of products here because that's what I want to return so I'll say products equal to new at the least okay and at the end I want to return this return products okay so let's let's run this and let's see what happens of course I want to print all the values so I will just use this for Loop which is already mentioned

05:29:24
here it's not copied copy paste okay this is not copy pting right this is code reuse what we say from prods prods yeah prods okay so basically let me just name this as a product because we're already using p somewhere we are using p here in fact you know what we can do we can also actually comment the entire section here so that we can focus only on with a product with a text okay so will this work let's try if I run this code of course you will not get any data here you know why we got empty

05:30:08
is because the list is empty we want to update this list from the all the products we have so we got all the products here from that product I want to fetch few products in fact you know I will just name this as prods here so that we can have different names prods okay so let's add here but how will you add of course we want to use a for Loop here because we want to iterate right uh so what I can do is I can use a for Loop and I want to iterate between the products I can say product p is equal to products so I want to iterate

05:30:44
between all the products and find and add only those products in this list which has a black in it it can be with a black in A Name Black in a type black in the uh place right so how do we do that so it's very simple you can simply use a if condition here and you can say p do get name do we have uh how do you do it can I use substring that's one option or if you talk about a string it also has one more method called contains you can see that we got contains because we don't want a string

05:31:24
we we just want to check if the text is there if it contains the word okay why is black here it should be uh text whatever text you're searching if that text is here or check in the place so I will say get or

maybe type check in a type does type contains this text or does my place contains this text contains the text so basically I'm checking everywhere I'm checking in the name I'm checking in the type I'm checking in the place as well right if it is there I will add this particular product in the prods so prods

05:32:09
do add p and of course you might have to rewatch this session right by yourself because when you're just watching on the go it's difficult to remember everything right so yeah before we go ahead hit that like button everyone come on sometime you have to keep YouTube Happy you know YouTube how YouTube promotes the video they basically see the number of likes they see they see the engagement they see the comments uh so do that let make YouTube Happy YouTube will make me happy and that's how it

05:32:40
works that's the EOS that's how the ecosystem works why okay i' okay there's a equal to here there should be a colon my bad okay cool uh I think this should work now let's see if this works I will just go back to my main or I can just stay stay here run okay still not working I think the problem is when you say black exactly this text black and if you see the products we don't have any black with small B you can see all our capital capital B Block B here okay that will not work so what we can do is we can

05:33:17
actually convert this into a lower case so whatever text we are getting I will simply say string s Str and so if you want to compare two strings it's always better to have in a lowercase so I will say text do to lowercase so and then don't compare with text compare with string because that's a lower case and not just this you have to also convert this G get name I me whatever get whatever name you get here whatever type you get here also into lower text so what you can do is you can

05:33:51
create some local variables here for the name in fact not here inside the loop because the value will change every time so I can say string name and this name will be P.G name and then we want a lower case of it so dot to lower case that's how you compare basically then we got string type is equal to P.G uh type to lowercase and again string uh place is equal to p. get place. to lowercase and now instead of comparing with the data which you get from the product just compare with the variables

05:34:36
which are already converted into a lower case you got the point right what is happening here and how do you solve that yeah so basically you have to convert it into a lower case or upper case it should be common case let's see if this works now yep it worked can see that we got all the products which has something black you can see the place is black black draw uh here it's also black table uh wcom is in Black drawer Logitech keyword is in black table uh HDMI cable is in Black draw Java black book blackbook then we

05:35:07
got black table black drawer and black Bas it worked so we are going to enhance this product more uh in terms of connecting to a database in terms of connecting to making this as a web application and then the way we have built this product we don't have to change much okay and the class this class will be used as it is you know you will see that once we start uh doing this okay what else we can do here so assignments for you let me just write down the assignments on my iPad so that you will remember the assignments uh

05:35:39
okay sure not Shing my iPad okay so assignment the first one you have to search by place okay that's one uh next is okay what what are the next thing you have to search by warranty search products out of warranty so the first assignment is search by play second is search product out of warranty next you need to implement a stream API here okay so if you if you see the uh the thing which you have done here let me show you that once again the thing which we have done here searching it with this this is basically uh a....give important and teach me like this only in very very simple

Nice, you've reached the **Product Management mini-project** part 🧃
I'll explain this **exactly like sir**, but in **super simple steps**, and I'll mark only the **important concepts** so you can revise fast.

🌼 1. Problem Story – What Are We Building?

Office example:

- There are many items: laptops, mouse, cables, etc.
- Problem: *"Yeh Asus laptop kahan hai?"*
  It may be on brown table, white table, black drawer, etc.
- Also we want to know:
  - Where is it kept? (place)
  - What type is it? (computer / cable / mouse / etc.)
  - Is it still under warranty?

So we need a **small console app** to:

- Store all products
- Show all products
- Search product by name
- Search products by text (like "black")
- (Assignments: by place, by out-of-warranty)

🌼 2. Thinking in Objects (Entity → Class → Object)

Sir says: "In Java, everything is object."

So we first design an **Entity: Product**

✨ Product has these properties:

- `name` – e.g. `"Asus VivoBook"`
- `type` – e.g. `"Laptop"`, `"Cable"`, `"Mouse"`

- `place` – e.g. `"Brown Table"`, `"Black Drawer"`
- `warranty` – e.g. `2022` (year until it is valid)

We convert this into a **Java class**:

```
java

public class Product {
    private String name;
    private String type;
    private String place;
    private int warranty;

    // 1️⃣ Default constructor
    public Product() {}

    // 2️⃣ Parameterized constructor
    public Product(String name, String type, String place, int warranty) {
        this.name = name;
        this.type = type;
        this.place = place;
        this.warranty = warranty;
    }

    // 3️⃣ Getters & setters (encapsulation)
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getType() { return type; }
    public void setType(String type) { this.type = type; }

    public String getPlace() { return place; }
    public void setPlace(String place) { this.place = place; }

    public int getWarranty() { return warranty; }
    public void setWarranty(int warranty) { this.warranty = warranty; }

    // 4️⃣ toString() — how to print object nicely
    @Override
    public String toString() {
        return "Product{" +
                "name='" + name + '\'' +
                ", type='" + type + '\'' +
                ", place='" + place + '\'' +
                ", warranty=" + warranty +
                '}';
    }
}
```

🔥 Important ideas here

- `private` fields + `get/set` = **encapsulation**
- `toString()` = how `System.out.println(product)` looks
- `Product` is your **entity**, just like `Alien` earlier in JPA code

🌼 3. Why not use `name1`, `name2`, `name3` etc.? (BAD way)

Initially we could do:

```java
String name1 = "Asus VivoBook";
String type1 = "Laptop";
String place1 = "Brown Table";
int warranty1 = 2022;

String name2 = "Apple Mouse";
String type2 = "Mouse";
String place2 = "White Table";
int warranty2 = 2025;
```

❌ Problems:

- For every new product: 4 more variables
- Code becomes ugly and hard to manage
- Searching & filtering are painful

✅ Better: **one `Product` class**, and many **`Product`** objects.

🌼 4. Delegation – Why `ProductService` Class?

Sir used the real-life example:

- Manager doesn't do everything.
- Work is **delegated** to team members.

Similarly:

- **`main()` should NOT** contain all logic.
- **`main()`** just **calls** methods.
- Actual logic goes into **`ProductService`** class.

So we create:

```java
public class ProductService {

    private List<Product> products = new ArrayList<>();

    // Add product to the list
    public void addProduct(Product p) {
        products.add(p);
    }

    // Add initial sample products
    public void addSomeProducts() {
        addProduct(new Product("Type C Cable", "Cable", "Black Drawer", 2024));
        addProduct(new Product("Mac Studio", "Computer", "White Table", 2026));
        addProduct(new Product("Asus VivoBook", "Laptop", "Brown Drawer", 2022));
        addProduct(new Product("Asus ROG", "Laptop", "Black Table", 2025));
        addProduct(new Product("Logitech Mouse", "Mouse", "Black Table", 2023));
        addProduct(new Product("Apple Mouse", "Mouse", "White Table", 2024));
        addProduct(new Product("Black Beast", "Computer", "White Table", 2025));
        // ...etc
    }
    // 1️⃣ Get all products
    public List<Product> getAllProducts() {
        return products;
    }

    // 2️⃣ Get product by exact name
    public Product getProductByName(String name) {
        for (Product p : products) {
            if (p.getName().equals(name)) {
                return p;
            }
        }
        return null; // not found
    }

    // 3️⃣ Search by text: name/type/place contains text
    public List<Product> getProductsWithText(String text) {
        List<Product> result = new ArrayList<>();

        String searchText = text.toLowerCase();

        for (Product p : products) {
            String name = p.getName().toLowerCase();
            String type = p.getType().toLowerCase();
            String place = p.getPlace().toLowerCase();

            if (name.contains(searchText) ||
                type.contains(searchText) ||
                place.contains(searchText)) {
                result.add(p);
            }
        }

        return result;
    }

    // 4️⃣ Assignment 1: search by place
    public List<Product> getProductsByPlace(String place) {
        List<Product> result = new ArrayList<>();
        String target = place.toLowerCase();

        for (Product p : products) {
            if (p.getPlace().toLowerCase().equals(target)) {
                result.add(p);
            }
        }
        return result;
    }

    // 5️⃣ Assignment 2: products out of warranty
    public List<Product> getOutOfWarranty(int currentYear) {
        List<Product> result = new ArrayList<>();

        for (Product p : products) {
            if (p.getWarranty() < currentYear) {
                result.add(p);
            }
        }
        return result;
    }
}
```

🔥 Important ideas here

- `List<Product>` = our **temporary in-memory database**
- `addSomeProducts()` = seeding sample data
- All searching logic is inside `ProductService` (delegation)
- Searching approach = simple **linear search** using `for` loop

🌼 5. Main Class – Talking to the Service

`main()` is like the **user**: it just asks for things.

```java
java

public class Main {
    public static void main(String[] args) {

        ProductService service = new ProductService();
        service.addSomeProducts();

        // 1️⃣ Print all products
        System.out.println("✅ All Products:");
        List<Product> all = service.getAllProducts();
        for (Product p : all) {
            System.out.println(p);
        }

        System.out.println("---------------");

        // 2️⃣ Get one product by exact name
        System.out.println("✅ Single Product (by name):");
        Product mouse = service.getProductByName("Logitech Mouse");
        System.out.println(mouse);

        System.out.println("---------------");

        // 3️⃣ Get products that contain text "black"
        System.out.println("✅ Products containing text 'black':");
        List<Product> blackItems = service.getProductsWithText("black");
        for (Product p : blackItems) {
            System.out.println(p);
        }

        System.out.println("---------------");

        // 4️⃣ Assignment demo: Products on 'White Table'
        System.out.println("✅ Products at place 'White Table':");
        List<Product> whiteTable = service.getProductsByPlace("White Table");
        for (Product p : whiteTable) {
            System.out.println(p);
        }

        System.out.println("---------------");

        // 5️⃣ Assignment demo: Out of warranty (assume current year 2025)
        System.out.println("✅ Out-of-warranty products (before 2025):");
        List<Product> outOfWarranty = service.getOutOfWarranty(2025);
        for (Product p : outOfWarranty) {
            System.out.println(p);
        }
    }
}
```

🌼 6. Key Java Concepts You Touched Here

Let me summarise all the **important Java concepts** hidden in this video:

1. **Class & Object**
    - **`Product`** class → many **`Product`** objects
2. **Encapsulation**

  - `private` fields
  - getters & setters
3. **Constructors**
  - Default constructor → `new Product()`
  - Parameterized constructor → `new Product("Asus", "Laptop", ... )`
4. **toString()**
  - For printing objects nicely
5. **List & ArrayList**
  - `List<Product> products = new ArrayList<>();`
  - acts like a simple database
6. **Service Layer / Delegation**
  - `ProductService` contains logic
  - `Main` just calls methods
7. **Enhanced for loop**

```java
for (Product p : products) {
    System.out.println(p);
}
```

8. **String functions**
  - `equals()` – for exact matching
  - `contains()` – for partial matching
  - `toLowerCase()` – for case-insensitive search
9. **Search logic**
  - Simple **linear search** over list

🌼 7. What will happen next (as sir hinted)

Later, this same idea will be extended to:

1. **Add Database (Hibernate / JPA + MySQL)**
  - Instead of `List<Product>`, store in table.
2. **Convert to Web Application**
  - Use **Spring Boot + REST API**
  - Endpoints like:
    - `GET /products`
    - `GET /products/{name}`
    - `GET /products/search?text=black`

- Same logic, just called via **HTTP** instead of `main()`.

So this small console project is actually the **base thinking** for real backend projects.

If you want, next I can:

- 🧠 Convert this Product app into **Spring Boot REST API** step-by-step,
  or
- 🎯 Explain **only Streams version** of these methods (same logic using Java 8 streams),
  or
- 📄 Make an **interview-style explanation** of this mini-project so you can tell it in viva / interview.

Just tell me:
"Next, explain Streams" or **"Next, convert this to Spring Boot REST"** or anything you want.

05:36:29
normal noral iteration right I want you to convert this into a stream API convert this code into stream API how will you do that so on my channel uh there is a video for stream API watch it you'll understand how to use stream API and Implement that so that's your third assignment so this is the work which you have to complete before the next session yeah and I I I love to see your pose by by saying this is all about learning it's not about MacBook and that's what what that's what we

05:37:01
want okay because see in your life if you know if you know something if you master something and if you're consistent you will get thousands of MacBooks it's not just about one okay it's of no use if you get a if you get a Macbook without knowledge and I have learned that in a hard way so basically in my entire career I've never focused on how much money I earn how many gadgets I have which car I drive it's more about can I what what I'm learning every day if I don't learn something every day today I

05:37:30
feel I'm not done enough okay so do that okay focus on learning not on the things cool so what's the agenda for today the agenda is very simple uh we are working on a project which is a product uh so basically we are managing the products which I have in my office and for that we were able to add the products in the list and then we were able to fetch it we were able to search it and I have seen your assignments uh you know I don't know I never thought I can make such a bigger impact in just it all

05:38:04
started with a small thought of giving giving away then we got a thought hey can we just do a challenge of 10 days and now looking at your response on different platforms is amazing uh every time I open LinkedIn now I get tons of uh notifications and then in the entire day I'm actually spending now two to three hours just checking your messages and uh notifications in uh in LinkedIn and that's quite fun so it was it was an amazing experience uh looking at your uh messages and notifications and one thing

05:38:43
you can do if you want to improve your rich on LinkedIn because it's a very good platform to con create networks create connections uh getting a job so what you can do is uh when you're are posting something try to make it more interesting uh maybe by putting an image of what you're doing example I saw one of some of the post uh let's say if you talk about Pascal triangle so having a pascal triangle in your LinkedIn post as a photo will give you a better reach because when people scroll on LinkedIn

05:39:12
they get excited looking at the images not the just the post and when you start with your post or the article or not the article but the post on LinkedIn uh the first line is very important that's how you can hook users there or your connections okay so let's talk about what we are going to do today it's very simple basically we are going to focus on a project which we are doing and then we'll convert that project into spring project now we will introduce a framework you know let me give you my

05:39:41
experience when I started my programming journey and then when I looked at uh different uh uh Frameworks in fact I started my career with Java then I worked on Spring swing uh tool which is the GUI Tool uh then I moved towards uh web development Sate and jsps and I was very happy but then someone suggested hey you know instead of working on sub JSP should work on STS framework and when the when I when first time when I heard about a framework or a tool I was like I I'm happy you know it's it's all

05:28:11
prods and I will say hey service get uh products with text and I will mention the text here so text is what I want to search based on a word like black uh okay but unfortunately we don't have this uh this word right so what do you think what can be done here okay I can get some yeah good good job so there there's option of substring we can do that uh yeah we can convert it into upper case and lower case okay we'll do that so what I will do is I don't have that method so let me click here in fact

05:28:48
you can also click here you can say create method get product with text and you can see we got the method here but at this point the method is not doing anything so basically I want list of products here because that's what I want to return so I'll say products equal to new at the least okay and at the end I want to return this return products okay so let's let's run this and let's see what happens of course I want to print all the values so I will just use this for Loop which is already mentioned

05:29:24
here it's not copied copy paste okay this is not copy pting right this is code reuse what we say from prods prods yeah prods okay so basically let me just name this as a product because we're already using p somewhere we are using p here in fact you know what we can do we can also actually comment the entire section here so that we can focus only on with a product with a text okay so will this work let's try if I run this code of course you will not get any data here you know why we got empty

05:30:08
is because the list is empty we want to update this list from the all the products we have so we got all the products here from that product I want to fetch few products in fact you know I will just name this as prods here so that we can have different names prods okay so let's add here but how

will you add of course we want to use a for Loop here because we want to iterate right uh so what I can do is I can use a for Loop and I want to iterate between the products I can say product p is equal to products so I want to iterate

05:30:44
between all the products and find and add only those products in this list which has a black in it it can be with a black in A Name Black in a type black in the uh place right so how do we do that so it's very simple you can simply use a if condition here and you can say p do get name do we have uh how do you do it can I use substring that's one option or if you talk about a string it also has one more method called contains you can see that we got contains because we don't want a string

05:31:24
we we just want to check if the text is there if it contains the word okay why is black here it should be uh text whatever text you're searching if that text is here or check in the place so I will say get or maybe type check in a type does type contains this text or does my place contains this text contains the text so basically I'm checking everywhere I'm checking in the name I'm checking in the type I'm checking in the place as well right if it is there I will add this particular product in the prods so prods

05:32:09
do add p and of course you might have to rewatch this session right by yourself because when you're just watching on the go it's difficult to remember everything right so yeah before we go ahead hit that like button everyone come on sometime you have to keep YouTube Happy you know YouTube how YouTube promotes the video they basically see the number of likes they see they see the engagement they see the comments uh so do that let make YouTube Happy YouTube will make me happy and that's how it

05:32:40
works that's the EOS that's how the ecosystem works why okay i' okay there's a equal to here there should be a colon my bad okay cool uh I think this should work now let's see if this works I will just go back to my main or I can just stay stay here run okay still not working I think the problem is when you say black exactly this text black and if you see the products we don't have any black with small B you can see all our capital capital B Block B here okay that will not work so what we can do is we can

05:33:17
actually convert this into a lower case so whatever text we are getting I will simply say string s Str and so if you want to compare two strings it's always better to have in a lowercase so I will say text do to lowercase so and then don't compare with text compare with string because that's a lower case and not just this you have to also convert this G get name I me whatever get whatever name you get here whatever type you get here also into lower text so what you can do is you can

05:33:51
create some local variables here for the name in fact not here inside the loop because the value will change every time so I can say string name and this name will be P.G name and then we want a lower case of it so dot to lower case that's how you compare basically then we got string type is equal to P.G uh type to lowercase and again string uh place is equal to p. get place. to lowercase and now instead of comparing with the data which you get from the product just compare with the variables

05:34:36

which are already converted into a lower case you got the point right what is happening here and how do you solve that yeah so basically you have to convert it into a lower case or upper case it should be common case let's see if this works now yep it worked can see that we got all the products which has something black you can see the place is black black draw uh here it's also black table uh wcom is in Black drawer Logitech keyword is in black table uh HDMI cable is in Black draw Java black book blackbook then we

05:35:07
got black table black drawer and black Bas it worked so we are going to enhance this product more uh in terms of connecting to a database in terms of connecting to making this as a web application and then the way we have built this product we don't have to change much okay and the class this class will be used as it is you know you will see that once we start uh doing this okay what else we can do here so assignments for you let me just write down the assignments on my iPad so that you will remember the assignments uh

05:35:39
okay sure not Shing my iPad okay so assignment the first one you have to search by place okay that's one uh next is okay what what are the next thing you have to search by warranty search products out of warranty so the first assignment is search by play second is search product out of warranty next you need to implement a stream API here okay so if you if you see the uh the thing which you have done here let me show you that once again the thing which we have done here searching it with this this is basically uh a

05:36:29
normal noral iteration right I want you to convert this into a stream API convert this code into stream API how will you do that so on my channel uh there is a video for stream API watch it you'll understand how to use stream API and Implement that so that's your third assignment so this is the work which you have to complete before the next session yeah and I I I love to see your pose by by saying this is all about learning it's not about MacBook and that's what what that's what we

05:37:01
want okay because see in your life if you know if you know something if you master something and if you're consistent you will get thousands of MacBooks it's not just about one okay it's of no use if you get a if you get a Macbook without knowledge and I have learned that in a hard way so basically in my entire career I've never focused on how much money I earn how many gadgets I have which car I drive it's more about can I what what I'm learning every day if I don't learn something every day today I

05:37:30
feel I'm not done enough okay so do that okay focus on learning not on the things cool so what's the agenda for today the agenda is very simple uh we are working on a project which is a product uh so basically we are managing the products which I have in my office and for that we were able to add the products in the list and then we were able to fetch it we were able to search it and I have seen your assignments uh you know I don't know I never thought I can make such a bigger impact in just it all

05:38:04
started with a small thought of giving giving away then we got a thought hey can we just do a challenge of 10 days and now looking at your response on different platforms is amazing uh every time I open LinkedIn now I get tons of uh notifications and then in the entire day I'm actually spending now two to three hours just checking your messages and uh notifications in uh in

LinkedIn and that's quite fun so it was it was an amazing experience uh looking at your uh messages and notifications and one thing

05:38:43
you can do if you want to improve your rich on LinkedIn because it's a very good platform to con create networks create connections uh getting a job so what you can do is uh when you're are posting something try to make it more interesting uh maybe by putting an image of what you're doing example I saw one of some of the post uh let's say if you talk about Pascal triangle so having a pascal triangle in your LinkedIn post as a photo will give you a better reach because when people scroll on LinkedIn

05:39:12
they get excited looking at the images not the just the post and when you start with your post or the article or not the article but the post on LinkedIn uh the first line is very important that's how you can hook users there or your connections okay so let's talk about what we are going to do today it's very simple basically we are going to focus on a project which we are doing and then we'll convert that project into spring project now we will introduce a framework you know let me give you my

05:39:41
experience when I started my programming journey and then when I looked at uh different uh uh Frameworks in fact I started my career with Java then I worked on Spring swing uh tool which is the GUI Tool uh then I moved towards uh web development Sate and jsps and I was very happy but then someone suggested hey you know instead of working on sub JSP should work on STS framework and when the when I when first time when I heard about a framework or a tool I was like I I'm happy you know it's it's all

05:40:10
working here why you have to learn a different framework to do the same thing of course the project which you build in without using a framework is same as when you when you build a project using a framework it's almost same right the output will be almost same then why to learn a new framework and do it it's because framework will reduce your work of course of course you have to learn something but with less efforts you will get more and that's what you get from the Frameworks and today after this

05:40:37
session I'm sure you're not going to learn spring framework because it's very difficult to teach spring framework in one hour but you will realize the power of spring you know lot of magical things are happening behind the scene and you will love it okay so yeah you're going to learn about the postgis as well which is a database I wanted to go for some uh database like myql but then you might have on my skill I wanted to give you a new flavor and that's why we have P okay so let me share my screen and let's try

05:41:07
to understand what is a problem here so this is the this is the this is what we are building here right so if you can see uh we have our code which is the main file uh in which you have basically what we have is we want to add products what now what is a product here so if you talk about a product class it has uh four uh private variables we have name type place and warranty uh example let's say a product can be anything example here if you see we have a product which is Type C cable which is a type of cable

05:41:38
uh and place his black drawer this is the warranty I'm not sure if you do really okay we don't get the warranty for uh cables but it's it's just that uh then we got Mac Studio which is a computer on white table and that's a warranty so basically what we did is we created different products and we have added all the products in the list and for that we have used a product service class and then uh in the product service class we have a list of elements here list of products and then we can add the

05:42:07
product we can uh get a single product by a name we can get the product by a text we can okay we I have also given you some assignment on warranty and stuff I hope you have done that okay this this is what we have done right and we were able to add the products we were able to fetch the products okay but then what is a problem we we are trying to solve here so before that let's understand the architecture of the project which we already have uh so we have a main file right so this is this is your main file and then uh if you

05:42:40
want to add a product basically we have a product service class which is doing all the logical part so from the main basically we are saying add product right but then how exactly product will be added the logic is maintained here so logic is there inside the product service now how to uh get the product so again we are just calling it right from the main we are just calling it and the logic is mentioned in the uh product service and then how do you represent a single product we don't want to use variables

05:43:13
and that's why we are uh basically using uh a class called Product now this product class represents an entity for you right now this will have a name and name Place type and warranty and basically in the product service what you have is now since we have multiple products we have to store it somewhere and where we are storing them is in the list so basically you have a list and inside this list you are storing all the products the problem with this is it's a very common problem where uh every time

05:43:42
you run this application so whatever application you have here every time you run this basically you are creating a new list and then you are initializing the list with this values and then you are fetching value right I don't want to do this I don't want to uh every time I run the application it should add the list like this ultimately we are working with a code right example if you have a new product basically we have to change the code here we don't want to do that so where do where do you store data

05:44:11
because at the moment I close this application I will lose all the data right and to store this data we have to store it somewhere and that's why uh in this particular project what we're going to do is we will be using a database here okay now this is your database and currently we are using postgress uh SQL which is one one of the most amazing open source project open source database okay I think you can you cannot see that that's yeah basically we are continuing with the EXT code so we

05:44:42
will be expanding it yeah and this is what I was talking about uh Wades has mentioned an important thing here uh which is code looks difficult uh yesterday but after doing it by myself it seems easy now that's that's important you have to try The Code by yourself because when I type I

have a clear picture in my mind because I'm typing it I know what how the flow going but when you watch a video or when you watch it live uh sometime you're not clear what is happening and that's why we have to

05:45:11
write The Code by yourself so basically what we are doing is we are creating a database now and we want to store this data in the database so basically whatever list you had here uh this list this list instead of storing it in a least we have to store that in the database directly now we can directly write the code so question arise how do you you write a code for database some we have to use something called J DBC now jdbc stands for Java database connectivity right so we need to use JBC here now the problem is if you want to

05:45:44
use jdbc we have to write some steps to work with jdbc and the question is where you will write the steps and that's where we have to use something called a layer so don't don't write your Dy code inside service service is meant for logic processing logic if you want to work with database what you can do is you can create another class here called Product uh product DB now this class is responsible to communicate with the database cool so that's what we are going to implement here I just need to check if everything

05:46:20
is proper in my machine just a second everyone yeah basically we have to get a different class right so what I'm going to do is in this code basically uh we are going to write database connectivity database connections and to achieve that uh we have to make some changes now what kind of changes we want here uh the changes are very simple if you want to connect to a database of course first of all we need a database here right now if you want to work with database what you can do is uh first of all we start with jdbc

05:46:47
part uh if you go back to okay so in my machine I have installed something called a postgress so you have to install postgress first now how do you install postgress very simple uh so go to browser and search for pogress this is where you can download it and pogress and mql is almost similar the difference is po is being open source now and Myas is a part of Oracle it's also open source I I guess there's one version of Myas which is open source I'm not sure but uh if you totally want to use the

05:47:18
the um the open source version I would prefer to go for poquiz and you can see you will also get the source code of it not needed but anyway so you can download the post in your machine and uh once you install it it's very simple to install it you just have to set the password you will get something called uh PG admin so once you install postgis you will get PG admin which is postgis admin and you can use myql if you want in your assignment uh so you have to set the password I have set the password

05:47:48
which is 000000 so four time0 and click on okay now this is how your post gist database will look like or post gist admin will look like and in this you will have an option of server so if you expand server okay just asking me for the password enter yeah so this is how your poin will look like I know this looks some fancy tool you know you can see all the dashboard uh the benchmarks what is happening with your system uh and if you can see we have databases here by default you will get this post

05:48:19
database test is what I have created uh but you can use post G and if you want to create a database which we are going to use what we can do is we can right click here and we can say create database datase and what database name I want to have here is teliscope so that's my database name if you want to see the SQL query for this you can click on SQL and that's the query so create database telescope uh we don't need all these things but then owner is postgress doesn't matter click on Save

05:48:48
and you can see you got your database here so this is your postgress the database which you created here and in this if you look at schemas now this is where your table you'll having your tables uh so I'm assuming that you know the concept of uh SQL database tables okay otherwise we have to take next six months to talk about SQL it's a huge topic but yeah we got we got the database with teliscope and here you can see we have tables and by default it is empty okay I want to create one table

05:49:19
here so what you can do is you can right click here and you can click on create table and what table I want so my table should be a product table okay so we want a product table and in this we want to okay so we want a product table and how many columns you want so basically you can just click on the column here and you can create the columns so the First Column which I want is ID so I know we don't have the ID in our remember when we started the when we started talking about our project of

05:49:54
product management I've mentioned that your product is actually an entity right so every entity will have one table so let's say in your project you have five entities every entity will have a table right uh so in the table basically you need to have your uh primary key so every table will have a primary key so in this case I'm creting a primary key here see name cannot be a primary key because I can have two laptops with the same name yeah possible right okay so I have ID uh what is a type of it I want

05:40:10
working here why you have to learn a different framework to do the same thing of course the project which you build in without using a framework is same as when you when you build a project using a framework it's almost same right the output will be almost same then why to learn a new framework and do it it's because framework will reduce your work of course of course you have to learn something but with less efforts you will get more and that's what you get from the Frameworks and today after this

05:40:37
session I'm sure you're not going to learn spring framework because it's very difficult to teach spring framework in one hour but you will realize the power of spring you know lot of magical things are happening behind the scene and you will love it okay so yeah you're going to learn about the postgis as well which is a database I wanted to go for some uh database like myql but then you might have on my skill I wanted to give you a new flavor and that's why we have P okay so let me share my screen and let's try

05:41:07
to understand what is a problem here so this is the this is the this is what we are building here

right so if you can see uh we have our code which is the main file uh in which you have basically what we have is we want to add products what now what is a product here so if you talk about a product class it has uh four uh private variables we have name type place and warranty uh example let's say a product can be anything example here if you see we have a product which is Type C cable which is a type of cable

05:41:38
uh and place his black drawer this is the warranty I'm not sure if you do really okay we don't get the warranty for uh cables but it's it's just that uh then we got Mac Studio which is a computer on white table and that's a warranty so basically what we did is we created different products and we have added all the products in the list and for that we have used a product service class and then uh in the product service class we have a list of elements here list of products and then we can add the

05:42:07
product we can uh get a single product by a name we can get the product by a text we can okay we I have also given you some assignment on warranty and stuff I hope you have done that okay this this is what we have done right and we were able to add the products we were able to fetch the products okay but then what is a problem we we are trying to solve here so before that let's understand the architecture of the project which we already have uh so we have a main file right so this is this is your main file and then uh if you

05:42:40
want to add a product basically we have a product service class which is doing all the logical part so from the main basically we are saying add product right but then how exactly product will be added the logic is maintained here so logic is there inside the product service now how to uh get the product so again we are just calling it right from the main we are just calling it and the logic is mentioned in the uh product service and then how do you represent a single product we don't want to use variables

05:43:13
and that's why we are uh basically using uh a class called Product now this product class represents an entity for you right now this will have a name and name Place type and warranty and basically in the product service what you have is now since we have multiple products we have to store it somewhere and where we are storing them is in the list so basically you have a list and inside this list you are storing all the products the problem with this is it's a very common problem where uh every time

05:43:42
you run this application so whatever application you have here every time you run this basically you are creating a new list and then you are initializing the list with this values and then you are fetching value right I don't want to do this I don't want to uh every time I run the application it should add the list like this ultimately we are working with a code right example if you have a new product basically we have to change the code here we don't want to do that so where do where do you store data

05:44:11
because at the moment I close this application I will lose all the data right and to store this data we have to store it somewhere and that's why uh in this particular project what we're going to do is we will be using a database here okay now this is your database and currently we are using postgress uh SQL which is one one of the most amazing open source project open source database okay I

think you can you cannot see that that's yeah basically we are continuing with the EXT code so we

05:44:42
will be expanding it yeah and this is what I was talking about uh Wades has mentioned an important thing here uh which is code looks difficult uh yesterday but after doing it by myself it seems easy now that's that's important you have to try The Code by yourself because when I type I have a clear picture in my mind because I'm typing it I know what how the flow going but when you watch a video or when you watch it live uh sometime you're not clear what is happening and that's why we have to

05:45:11
write The Code by yourself so basically what we are doing is we are creating a database now and we want to store this data in the database so basically whatever list you had here uh this list this list instead of storing it in a least we have to store that in the database directly now we can directly write the code so question arise how do you you write a code for database some we have to use something called J DBC now jdbc stands for Java database connectivity right so we need to use JBC here now the problem is if you want to

05:45:44
use jdbc we have to write some steps to work with jdbc and the question is where you will write the steps and that's where we have to use something called a layer so don't don't write your Dy code inside service service is meant for logic processing logic if you want to work with database what you can do is you can create another class here called Product uh product DB now this class is responsible to communicate with the database cool so that's what we are going to implement here I just need to check if everything

05:46:20
is proper in my machine just a second everyone yeah basically we have to get a different class right so what I'm going to do is in this code basically uh we are going to write database connectivity database connections and to achieve that uh we have to make some changes now what kind of changes we want here uh the changes are very simple if you want to connect to a database of course first of all we need a database here right now if you want to work with database what you can do is uh first of all we start with jdbc

05:46:47
part uh if you go back to okay so in my machine I have installed something called a postgress so you have to install postgress first now how do you install postgress very simple uh so go to browser and search for pogress this is where you can download it and pogress and mql is almost similar the difference is po is being open source now and Myas is a part of Oracle it's also open source I I guess there's one version of Myas which is open source I'm not sure but uh if you totally want to use the

05:47:18
the um the open source version I would prefer to go for poquiz and you can see you will also get the source code of it not needed but anyway so you can download the post in your machine and uh once you install it it's very simple to install it you just have to set the password you will get something called uh PG admin so once you install postgis you will get PG admin which is postgis admin and you can use myql if you want in your assignment uh so you have to set the password I have set the password

05:47:48

which is 000000 so four time0 and click on okay now this is how your post gist database will look like or post gist admin will look like and in this you will have an option of server so if you expand server okay just asking me for the password enter yeah so this is how your poin will look like I know this looks some fancy tool you know you can see all the dashboard uh the benchmarks what is happening with your system uh and if you can see we have databases here by default you will get this post

05:48:19
database test is what I have created uh but you can use post G and if you want to create a database which we are going to use what we can do is we can right click here and we can say create database datase and what database name I want to have here is teliscope so that's my database name if you want to see the SQL query for this you can click on SQL and that's the query so create database telescope uh we don't need all these things but then owner is postgress doesn't matter click on Save

05:48:48
and you can see you got your database here so this is your postgress the database which you created here and in this if you look at schemas now this is where your table you'll having your tables uh so I'm assuming that you know the concept of uh SQL database tables okay otherwise we have to take next six months to talk about SQL it's a huge topic but yeah we got we got the database with teliscope and here you can see we have tables and by default it is empty okay I want to create one table

05:49:19
here so what you can do is you can right click here and you can click on create table and what table I want so my table should be a product table okay so we want a product table and in this we want to okay so we want a product table and how many columns you want so basically you can just click on the column here and you can create the columns so the First Column which I want is ID so I know we don't have the ID in our remember when we started the when we started talking about our project of

05:49:54
product management I've mentioned that your product is actually an entity right so every entity will have one table so let's say in your project you have five entities every entity will have a table right uh so in the table basically you need to have your uh primary key so every table will have a primary key so in this case I'm creting a primary key here see name cannot be a primary key because I can have two laptops with the same name yeah possible right okay so I have ID uh what is a type of it I want

05:50:23
the type to be uh integer do we have integer here yeah we want integer of type lens scale doesn't matter this should be a primary key I also want this to be uh to have autogenerated values I don't want to generate value by myself so every time I enter a new product it should give it a new ID 1 2 3 and the way you can do that is I'm using pus after a long time so there should be an option of uh identity generated no identity identity always increment by one start by one and that's it I think it should work

05:51:11
let's see the SQL query for this uh generated as always as entity uh identity okay increment one start one cool if I click on Save done so you can see we got table here but I want to check if this this table works what I can do is I can just go to the query tool where I can write the SQL queries so this

is a place where you can write SQL query I know I'm going bit fast but you have to rewatch this session this this is not about teaching you uh SQL this is more about how do use how do we

05:51:44
build a project right so what I will do is I will first of all say select star from product let's see if you have any product there uh it's empty uh relation product does not exist is it I created this table is it case sensitive from when no something is not working so I'll say use the lisco uh select star from product is okay something is not working what what's wrong I've clicked on this I'm using this database let me just refresh it once and run okay something is weird how about this if I say this

05:52:45
belongs to delisco this is not working columns is there okay we got only one column oh I forgot to enter all the different columns here you know what I will do I will just drop this and let's create this table by query so I will say create table product uh the product should have an ID and if you want to have a SQL query you can also mention uh serial okay you have to mention this of type integer serial if you want the autogenerated and it should be a primary key and as dbms changes this also this

05:53:42
also changes yeah now table is not Tesco database is Tesco and yeah I'm in the right thing doesn't matter and then we can say name is name is text I think uh po supports text not string or vcad so we got name uh then we got type which is also a text then we got place which is also a text text and then we got what we have one more thing which is warranty yeah so warranty which is of type integer let's run this and let's see what happens okay did I mention the wrong query do don't have to mention integer is it

05:54:33
so it says near integer okay yeah it worked okay so we should not mention integer if you mentioning cial because it is uh Cal is also a type okay cool uh we have created this table I will just say now let's say let's let's say if this work select star from uh product I I want this to be empty yeah you can say it is empty now so we got a table and this Auto increment now this will increment automatically I will not even insert this but then I want to do that from the Java code now so if you

05:55:06
can see I'm adding a product here but if you see we have we are adding all the products and then in the service we are adding in the list but what we are going to do is we'll not be doing the changes in this particular file the problem is when you are working with a database connections we need to create a different uh we have to add some libraries we need to connect with jdbc right for jdbc we have to add a jar file okay and then you can you can actually do that in the external files you can

05:55:35
just go back to external files and can add or EXN libraries you can add a jar file but to better manage the product we can use something called Maven now Maven is a build tool which also helps you to handle the jar files so what I can do is I can right click here I hope there should be option of converting to Maven okay do we have it here refactor now I can't see any option let me know in the chat if you know where to because I have used this on Eclipse because in Eclipse you get this option here no

05:56:07
problem what we can do is uh we can simply create a new project and let's say new project okay

and then instead of selecting a normal project what we're going to do is we're going to select a MAV project and we'll name this as product demo DB because we're working with with DB database now and then uh everything is as usual so okay there's a question m versus G which is uh which is best or which is better both are good it's just that I'm more comfortable with Maven because it

05:56:44
is easier to teach in Gradle there are lot of uh new things and then indentation is one thing so for projects grle is good for teaching MAV is good yeah and for beginners MAV is good Okay click on Create and it will create a m Pro project I will say new window because I want both these windows okay and then in this I'm going to Simply copy the files so if I go back here I want all these files so I can just simply copy and paste over it all and I'm also going to change so packages com do sorry

05:57:17
o. example let's stick to that now what you get in the maven project is this if you expand this or if if you go to pom.xml so whatever dependencies you need you just need to mention those dependencies here so what you can do is you can put a thing called dependencies and whatever dependency you want you can just add it here now what dependency we want we want a dependency for uh for this what is uh the connector right now how do you get the connector how do you get the dependency do you have do we

05:57:48
have to BU heart it uh not exactly so what you can do is go back to your browser there is a website called mvn repository and in this yeah so in this you just have to search for post GIS SQL jdbc and if you can see we have it here this is the library which we have to use and make sure that you get the recent ones okay don't go for the versions which are way old uh and which one to choose here so I I always go for second last version because all the latest version will always have some issues

05:58:19
okay and it also tells you which of the library has VAB abilities and you can see there are certain VAB abilities here this one is safe but we'll go for this one 42 . 5.4 select this one and copy this go back to your project and just paste so what you got there is dependency okay and every time you make the changes depending upon different IDs you will get different options yeah okay here add framework support Maven okay we got the option cool hey done oh it was so fast we don't even need to have a create a project so

05:58:55
what you do is you can just click on this right click on the project thank you so much Prine add framework support and select Maven here and you will get this option oh that was cool so now let's go back here specify the dependencies and paste it here okay so good thing is you don't have to create a new project thank you and now the moment you add this what you have to do is you have to refresh your Maven so if you can see there's the option of Maven changes click on this and it will download the jar files

05:59:24
for you if you want to see where it is click on the external libraries and you will find a jar file for po cool so this is what we wanted now once you have added that support it's very simple now what you do is instead of adding to our list we are not going to do that now so what I will do is I will just come in this section and in every way we have to make the changes okay let's not comment because our code is dependent on it we are not going to use it so let's say if I want to add a product so what do you

> 05:59:52
> do is instead of adding to a list because not we are not going to do that we're not going to add to a list we will add this to a database now how do you connect to a database it's very simple what you do is you simply say I want a database connection so I can say I can say DB dot uh DB dot add a product or maybe I can say save db. save and pass P just by writing this line the data should be stored in the database right will this work of course not right it's not Rajan to know everything we are not
>
> 06:00:27
> sure which database you're using uh what is your database name I mean which dbms you are using what is your database name what is the password to connect with it so this will not work so what we want is we want a extra layer remember in the iPad when I was talking about it we have mentioned about the extra layer this is the layer which we want to create so how do we create that layer it's very simple you can just right click on your project here and create a new class I name this class as product....give important and teach me like this only in very very simple

Samiksha... 🤍
I know this is a **huge** transcript.
You only want:

👉 **The important things**
👉 **Explained very very simple**
👉 **Same style as sir**
👉 **Specifically: Convert normal loop → Stream API version**

So I will give you **only the part you need**, in the same teaching tone.

🌼 PART 1 — What is the exact code we are converting?

Sir wrote this **normal iteration (enhanced for-loop)**:

```
java

List<Product> prods = new ArrayList<>();

for (Product p : products) {
    String name = p.getName().toLowerCase();
    String type = p.getType().toLowerCase();
    String place = p.getPlace().toLowerCase();

    if (name.contains(text) || type.contains(text) || place.contains(text)) {
        prods.add(p);
    }
}

return prods;
```

This is called:

✔️ Normal Iteration

✔️ Linear search
✔️ Manual filtering

🌼 PART 2 — Convert this to Stream API (VERY SIMPLE)

Streams allow you to write the same logic in **one line**, readable, simple.

🔥 Stream API version:

```java
return products.stream()
        .filter(p ->
            p.getName().toLowerCase().contains(text) ||
            p.getType().toLowerCase().contains(text) ||
            p.getPlace().toLowerCase().contains(text)
        )
        .collect(Collectors.toList());
```

That’s it.
Your normal loop → **converted into Stream API.**

🌼 PART 3 — Super Simple Explanation (Like Sir)

Think of **Stream** as:

👉 “Take the list…
👉 …pass it through a filter…
👉 …get only matching items.”

That's exactly what `.stream().filter().collect()` does.

🌼 PART 4 — Step-by-step breakdown (baby-level simple)

1️⃣ Start the stream

```java
products.stream()
```

Meaning:
"Start reading all products one-by-one automatically."

2️⃣ Apply filter (same logic as `if`)

```java
.filter(p -> p.getName().toLowerCase().contains(text))
```

Means:
"Keep only those products whose name contains the text."

We added 3 conditions:

```java
p.getName().toLowerCase().contains(text) ||
p.getType().toLowerCase().contains(text) ||
p.getPlace().toLowerCase().contains(text)
```

3️⃣ Collect back into a list

```java
.collect(Collectors.toList());
```

Means:
"Convert the results back into a List."

🌼 PART 5 — Full method in Stream API (clean version)

```java
public List<Product> getProductsWithText(String text) {
    String search = text.toLowerCase();

    return products.stream()
            .filter(p ->
                p.getName().toLowerCase().contains(search) ||
                p.getType().toLowerCase().contains(search) ||
                p.getPlace().toLowerCase().contains(search)
            )
            .collect(Collectors.toList());
}
```

🌼 PART 6 — Bonus: Other assignments in Stream API (very simple)

1️⃣ Search by place (Stream version)

```java
public List<Product> getProductsByPlace(String place) {
    String p = place.toLowerCase();

    return products.stream()
            .filter(prod -> prod.getPlace().toLowerCase().equals(p))
            .collect(Collectors.toList());
}
```

2️⃣ Products out of warranty (Stream version)

```java
public List<Product> getOutOfWarranty(int currentYear) {
    return products.stream()
```

```
            .filter(prod -> prod.getWarranty() < currentYear)
            .collect(Collectors.toList());
}
```

🌼 PART 7 — Why Stream API is better?

✔️ Cleaner
✔️ Less code
✔️ Easy to understand
✔️ No `new ArrayList<>()`
✔️ No manual looping
✔️ No `if` spam

Sir said:

> "This is code refactoring, not new logic. Same logic, cleaner code."

Exactly.

🌼 PART 8 — If you want, I can also explain:

- Stream API full concept (map, filter, reduce)
- Convert your **entire ProductService** into stream versions
- Explain with diagrams and flow
- Teach Stream API like you're a beginner
- Convert to **Spring Boot REST API** next

Just tell me:

👉 "Explain Stream API completely"
or
👉 "Convert full ProductService to streams"
or
👉 "Next, teach me Spring Boot conversion"

I will explain very very simple, like sir.

06:00:56
DB okay later and in this class basically you have to create a save method so I want the object of product DB so I will say product DB and I will say this is DB equal to new product DB once I get this object I can simply say db. save but unfortunately we don't have a database here so what I can do is or we don't have a save method so I will click here and say create method save and done we we got this method here and this method is responsible to store data in the database the entire service class should

06:01:36
not have any idea about database it will simply say save it I don't care how you do it it is your job prod DB to connect to database and save it now to connect with database you have to learn how do you use ddbc so you will find the desp in the description you will find some videos one is for spring framework one is for longbo which we're going to use and also for jdbc so in this we have to connect to database right and I will not be teaching you database connections jdbc connections here I'm just going to

06:02:06
write it so basically for databas connections you need something called a connection object so you have to say connection which is coming from java.sql now this is coming because you have added that jar file of pogress if you don't add this jar file you will not get this so we'll say connection con equal to okay we'll just say equal to null by default and and let's create this object in the Constructor so I will say public product DB Constructor and in the Constructor of this I will simply create

05:50:23
the type to be uh integer do we have integer here yeah we want integer of type lens scale doesn't matter this should be a primary key I also want this to be uh to have autogenerated values I don't want to generate value by myself so every time I enter a new product it should give it a new ID 1 2 3 and the way you can do that is I'm using pus after a long time so there should be an option of uh identity generated no identity identity always increment by one start by one and that's it I think it should work

05:51:11
let's see the SQL query for this uh generated as always as entity uh identity okay increment one start one cool if I click on Save done so you can see we got table here but I want to check if this this table works what I can do is I can just go to the query tool where I can write the SQL queries so this is a place where you can write SQL query I know I'm going bit fast but you have to rewatch this session this this is not about teaching you uh SQL this is more about how do use how do we

05:51:44
build a project right so what I will do is I will first of all say select star from product let's see if you have any product there uh it's empty uh relation product does not exist is it I created this table is it case sensitive from when no something is not working so I'll say use the lisco uh select star from product is okay something is not working what what's wrong I've clicked on this I'm using this database let me just refresh it once and run okay something is weird how about this if I say this

05:52:45
belongs to delisco this is not working columns is there okay we got only one column oh I forgot to enter all the different columns here you know what I will do I will just drop this and let's create this table by query so I will say create table product uh the product should have an ID and if you want to have a SQL query you can also mention uh serial okay you have to mention this of type integer serial if you want the autogenerated and it should be a primary key and as dbms changes this also this

05:53:42
also changes yeah now table is not Tesco database is Tesco and yeah I'm in the right thing doesn't matter and then we can say name is name is text I think uh po supports text not string or vcad so we got name uh then we got type which is also a text then we got place which is also a text text and then we got what we have one more thing which is warranty yeah so warranty which is of type

integer let's run this and let's see what happens okay did I mention the wrong query do don't have to mention integer is it

05:54:33
so it says near integer okay yeah it worked okay so we should not mention integer if you mentioning cial because it is uh Cal is also a type okay cool uh we have created this table I will just say now let's say let's let's say if this work select star from uh product I I want this to be empty yeah you can say it is empty now so we got a table and this Auto increment now this will increment automatically I will not even insert this but then I want to do that from the Java code now so if you

05:55:06
can see I'm adding a product here but if you see we have we are adding all the products and then in the service we are adding in the list but what we are going to do is we'll not be doing the changes in this particular file the problem is when you are working with a database connections we need to create a different uh we have to add some libraries we need to connect with jdbc right for jdbc we have to add a jar file okay and then you can you can actually do that in the external files you can

05:55:35
just go back to external files and can add or EXN libraries you can add a jar file but to better manage the product we can use something called Maven now Maven is a build tool which also helps you to handle the jar files so what I can do is I can right click here I hope there should be option of converting to Maven okay do we have it here refactor now I can't see any option let me know in the chat if you know where to because I have used this on Eclipse because in Eclipse you get this option here no

05:56:07
problem what we can do is uh we can simply create a new project and let's say new project okay and then instead of selecting a normal project what we're going to do is we're going to select a MAV project and we'll name this as product demo DB because we're working with with DB database now and then uh everything is as usual so okay there's a question m versus G which is uh which is best or which is better both are good it's just that I'm more comfortable with Maven because it

05:56:44
is easier to teach in Gradle there are lot of uh new things and then indentation is one thing so for projects grle is good for teaching MAV is good yeah and for beginners MAV is good Okay click on Create and it will create a m Pro project I will say new window because I want both these windows okay and then in this I'm going to Simply copy the files so if I go back here I want all these files so I can just simply copy and paste over it all and I'm also going to change so packages com do sorry

05:57:17
o. example let's stick to that now what you get in the maven project is this if you expand this or if if you go to pom.xml so whatever dependencies you need you just need to mention those dependencies here so what you can do is you can put a thing called dependencies and whatever dependency you want you can just add it here now what dependency we want we want a dependency for uh for this what is uh the connector right now how do you get the connector how do you get the dependency do you have do we

05:57:48
have to BU heart it uh not exactly so what you can do is go back to your browser there is a website

called mvn repository and in this yeah so in this you just have to search for post GIS SQL jdbc and if you can see we have it here this is the library which we have to use and make sure that you get the recent ones okay don't go for the versions which are way old uh and which one to choose here so I I always go for second last version because all the latest version will always have some issues

05:58:19
okay and it also tells you which of the library has VAB abilities and you can see there are certain VAB abilities here this one is safe but we'll go for this one 42 . 5.4 select this one and copy this go back to your project and just paste so what you got there is dependency okay and every time you make the changes depending upon different IDs you will get different options yeah okay here add framework support Maven okay we got the option cool hey done oh it was so fast we don't even need to have a create a project so

05:58:55
what you do is you can just click on this right click on the project thank you so much Prine add framework support and select Maven here and you will get this option oh that was cool so now let's go back here specify the dependencies and paste it here okay so good thing is you don't have to create a new project thank you and now the moment you add this what you have to do is you have to refresh your Maven so if you can see there's the option of Maven changes click on this and it will download the jar files

05:59:24
for you if you want to see where it is click on the external libraries and you will find a jar file for po cool so this is what we wanted now once you have added that support it's very simple now what you do is instead of adding to our list we are not going to do that now so what I will do is I will just come in this section and in every way we have to make the changes okay let's not comment because our code is dependent on it we are not going to use it so let's say if I want to add a product so what do you

05:59:52
do is instead of adding to a list because not we are not going to do that we're not going to add to a list we will add this to a database now how do you connect to a database it's very simple what you do is you simply say I want a database connection so I can say I can say DB dot uh DB dot add a product or maybe I can say save db. save and pass P just by writing this line the data should be stored in the database right will this work of course not right it's not Rajan to know everything we are not

06:00:27
sure which database you're using uh what is your database name I mean which dbms you are using what is your database name what is the password to connect with it so this will not work so what we want is we want a extra layer remember in the iPad when I was talking about it we have mentioned about the extra layer this is the layer which we want to create so how do we create that layer it's very simple you can just right click on your project here and create a new class I name this class as product

06:00:56
DB okay later and in this class basically you have to create a save method so I want the object of product DB so I will say product DB and I will say this is DB equal to new product DB once I get this object I can simply say db. save but unfortunately we don't have a database here so what I can do is or we don't have a save method so I will click here and say create method save and done we we got this method here and this method is responsible to store data in the database the entire

service class should

06:01:36
not have any idea about database it will simply say save it I don't care how you do it it is your job prod DB to connect to database and save it now to connect with database you have to learn how do you use ddbc so you will find the desp in the description you will find some videos one is for spring framework one is for longbo which we're going to use and also for jdbc so in this we have to connect to database right and I will not be teaching you database connections jdbc connections here I'm just going to

06:02:06
write it so basically for databas connections you need something called a connection object so you have to say connection which is coming from java.sql now this is coming because you have added that jar file of pogress if you don't add this jar file you will not get this so we'll say connection con equal to okay we'll just say equal to null by default and and let's create this object in the Constructor so I will say public product DB Constructor and in the Constructor of this I will simply create

06:02:36
the object and the way you do that is by using a driver manager. get connection and in this you have to mention the URL okay now it will take three parameters from you not one it will take three parameters the first parameter which you have to mention is about U the URL of the database the second one is a username and I know the username so this is po that's the username the password for my machine POG is 0 you when you install make sure that you change it and then we have a URL here now this is

06:03:11
something you have to remember or byart that's your choice but when you work from a long time you remember that so you say jdbc colon uh postgress SQL colon SL slash not this slash sl/ Local Host because it is installed on your machine colon 5432 which is the port number for postgress okay for different database we have different port number uh I think for MySQL oh I forgot the port of MySQL that's weird so this is the port of a postgress so you you got this detail right we got the URL we got the username

06:03:47
and password uh so you might get an exception what you can do is you can click on more actions and you can say surround with try catch because it might throw an exception so one once you got this database connection created let's save the product now to save it what you need to do is you have to write SQL query right so what we need SQL query here so I can say query now what will be my query my query is very simple I want to say insert into product uh with the fields because I don't want to mention the id id should

06:04:20
be autogenerated so I want to insert name I want to insert uh type of the product I want to insert the place I want to insert the warranty cool uh and then values we have to insert the values as well now question arise from where you will get the values tell me if I want to insert all this data from where you will get the values so basically the values you will get is from the product this is where you have all the values right because one product will have details about your product it will have a name

06:04:49

type place and warranty so what you can do is you can add the values here okay but since we are going to use prepare statement in jdbc now if you want to know more about jdbc you will find the link in description and you can watch the later video later about jdbc but here I will say question mark question mark question mark four question marks for four different values and we can replace that later but how do you execute the query so we have to create something called a prepared statement St equal to con do prepare

06:05:18
statement and we just have to pass the query here so whatever query you have created just pass it here now this might also through an exception so what I can do is I can say surround with try catch we are just creating a query now let's execute it so how do we execute it so we can simply say st. execute now what it will do is it will execute the query so let me just go go with the flow again in the main what we are doing is we are saying add product now this product is actually going to product service inside

06:05:51
product service we have this add product method which is taking a product it is asking database to save it so this same object goes to product DB and then you are accepting it here and we are using SQL query to send it to dat to the database right uh but the problem is we can't simply say execute because we have not mentioned the values yet it is still question mark we have to we have to change these values now how do we change it quite simple just before the execution you can say St dot each

06:06:21
question mark is one parameter right so you can say Set uh set string because the first one is string so the first question mark since we have four question marks we have to replace it the first question mark will be replaced by the name so p. get name the second question mark will be replaced by uh p. get type no yeah type the third question mark set string again the third question mark will be replaced by P do uh get place and last question mark which is set integer because last one is a integer here so it's a set int question

06:06:56
number four I mean the question mark 4 this is speed. get warranty so we are able to set all the values and we are saying execute our job is done the question is will this work I'm not sure let's try so what we can do is we can simply comment all other things only focus on adding a product what do you think will it work or did I completed all the steps I'm not sure sometime you get to know things when it gives you errors right uh so let's go go back here say run oh we got an error it says po uh phrase unable to

06:07:37
pass the phrase what's wrong what's wrong what's wrong what's wrong just check oh someone also mentioned that you can wrap it where do you have the option of wrapping it in the oh yeah okay this looks good so we can wrap the entire output cool so uh we we forgot to do one more thing we forgot to mention the database name so the database name is telis here okay we have not started with the data the data jpa yet I want you to show you uh the part of data jpa right and that's why I'm doing I'm going step

06:08:09
by step here let's try to run this once more time and oh something happened you can see there's no error it says process finished with exit okay let's try let's see how do you verify this let's open our PG admin and let's run run this particular query oh can you see that we got ID which is

autogenerated and we got all this data here in the database this looks cool okay now uh once we have entered the data let me commend the entire section because now we have data in the database so let's so let's let's not

06:08:48
insert every time now this time I want to fetch all the products this is something I want to execute okay so get all the products uh so here when you say get all products let's not get it from the list question from where you will get it again simple DB dot you can say uh get all this is the method I want so in the database I want to say get all but unfortunately we don't have a method so what you can do is you can say create method get all so we got the method and this should return a list so let me create a list here so I

06:09:24
will say list of products and we'll say products equals to new okay so people who are finding it difficult my suggestion would be First Watch the jdbc video and then then also watch the maven video so jdbc video and Maven video then watch the entire session you will love it you will see the power of Frameworks and how everything works okay so basically what we're doing is instead of getting data from a list last time we were getting data from the list right this time we want to fresh data from the database so

06:09:54
here we are saying we creating a list and let's fill this list from the values which is coming from database okay how do we do that so just say return product so that you will get rid of errors this time it will return empty let's let me try that let me see if I get the empty here let's run this and you can see we are not getting any data because it's empty the list is empty so how do we fill this it's very simple again use the same thing let's copy this because we are going to query the database paste

06:10:28
what are the changes the first change you're going to make is the query we're not going to insert this time we're going to select so we'll say select star uh okay we are not getting star we are going to mention the column name because we don't want ID there so we'll say name type place and warranty fromom table name is product and we don't have to apply well Clause is because we want all the data so basically we are fing everything from database right and and let's execute the query now since we

06:10:58
don't have any question mark here you can skip the entire section from here and simply say execute the difference is execute will not return the data we want data right so execute query is something you can use to to get the data and what it returns is the result set so it will return you a result set and what you can do is you can get data from this result set and store it in a list but the problem is it will give you uh d data in the result set format you have to convert that okay cool so basically we

06:11:31
want to F data from result set and set it in the products here because that's what we returning here right so how do you add so basically we can run a loop because if you say RS dot there's no method using which you can get all the data and save it in a list we have to do it manually one by one uh there's no option so what I will do is I will use a loop here and F data one by one so you can say fesh dat till the end and every time you fet data you can simply okay what we have to do is we have to create

06:12:02
a product object because okay I know we don't have time to explain this now but just go with the flow everyone so what we trying to do is we want to create a list of products right so we will be having multiple products so every time you run this Loop create a product object and it is highly recommended make sure that you watch the video of jdbc and Maven trust me this will be very very easy okay cool so uh let's get the product object and save each value in the product values okay uh so how do we

06:12:33
update the value so we have to say p do set name and this is coming from rs. getet string from the First Column so since we are getting multiple columns right this is the First Column second column third column fourth column then we can say p do Set uh type this is coming from second column so I will say get string column 2 p set place you know this is like a beautiful poem from a language which you don't understand right it sounds beautiful but you're not sure what is happening three then P do set warranty

06:13:11
and rs. get string or get integer this time it is integer and we are saying four okay so we were able to set the value for the product now once you get all the values just add the product in the product so we say products do add P now let's say you have have 15 values right so in the database now this while loop will run 15 times and every time you will get the new product object you are setting the values which is coming from database and adding to a list the same list goes back to the product

06:02:36
the object and the way you do that is by using a driver manager. get connection and in this you have to mention the URL okay now it will take three parameters from you not one it will take three parameters the first parameter which you have to mention is about U the URL of the database the second one is a username and I know the username so this is po that's the username the password for my machine POG is 0 you when you install make sure that you change it and then we have a URL here now this is

06:03:11
something you have to remember or byart that's your choice but when you work from a long time you remember that so you say jdbc colon uh postgress SQL colon SL slash not this slash sl/ Local Host because it is installed on your machine colon 5432 which is the port number for postgress okay for different database we have different port number uh I think for MySQL oh I forgot the port of MySQL that's weird so this is the port of a postgress so you you got this detail right we got the URL we got the username

06:03:47
and password uh so you might get an exception what you can do is you can click on more actions and you can say surround with try catch because it might throw an exception so one once you got this database connection created let's save the product now to save it what you need to do is you have to write SQL query right so what we need SQL query here so I can say query now what will be my query my query is very simple I want to say insert into product uh with the fields because I don't want to mention the id id should

06:04:20
be autogenerated so I want to insert name I want to insert uh type of the product I want to insert

the place I want to insert the warranty cool uh and then values we have to insert the values as well now question arise from where you will get the values tell me if I want to insert all this data from where you will get the values so basically the values you will get is from the product this is where you have all the values right because one product will have details about your product it will have a name

06:04:49
type place and warranty so what you can do is you can add the values here okay but since we are going to use prepare statement in jdbc now if you want to know more about jdbc you will find the link in description and you can watch the later video later about jdbc but here I will say question mark question mark question mark four question marks for four different values and we can replace that later but how do you execute the query so we have to create something called a prepared statement St equal to con do prepare

06:05:18
statement and we just have to pass the query here so whatever query you have created just pass it here now this might also through an exception so what I can do is I can say surround with try catch we are just creating a query now let's execute it so how do we execute it so we can simply say st. execute now what it will do is it will execute the query so let me just go go with the flow again in the main what we are doing is we are saying add product now this product is actually going to product service inside

06:05:51
product service we have this add product method which is taking a product it is asking database to save it so this same object goes to product DB and then you are accepting it here and we are using SQL query to send it to dat to the database right uh but the problem is we can't simply say execute because we have not mentioned the values yet it is still question mark we have to we have to change these values now how do we change it quite simple just before the execution you can say St dot each

06:06:21
question mark is one parameter right so you can say Set uh set string because the first one is string so the first question mark since we have four question marks we have to replace it the first question mark will be replaced by the name so p. get name the second question mark will be replaced by uh p. get type no yeah type the third question mark set string again the third question mark will be replaced by P do uh get place and last question mark which is set integer because last one is a integer here so it's a set int question

06:06:56
number four I mean the question mark 4 this is speed. get warranty so we are able to set all the values and we are saying execute our job is done the question is will this work I'm not sure let's try so what we can do is we can simply comment all other things only focus on adding a product what do you think will it work or did I completed all the steps I'm not sure sometime you get to know things when it gives you errors right uh so let's go go back here say run oh we got an error it says po uh phrase unable to

06:07:37
pass the phrase what's wrong what's wrong what's wrong what's wrong just check oh someone also mentioned that you can wrap it where do you have the option of wrapping it in the oh yeah okay this looks good so we can wrap the entire output cool so uh we we forgot to do one more thing we forgot to mention the database name so the database name is telis here okay we have

not started with the data the data jpa yet I want you to show you uh the part of data jpa right and that's why I'm doing I'm going step

06:08:09
by step here let's try to run this once more time and oh something happened you can see there's no error it says process finished with exit okay let's try let's see how do you verify this let's open our PG admin and let's run run this particular query oh can you see that we got ID which is autogenerated and we got all this data here in the database this looks cool okay now uh once we have entered the data let me commend the entire section because now we have data in the database so let's so let's let's not

06:08:48
insert every time now this time I want to fetch all the products this is something I want to execute okay so get all the products uh so here when you say get all products let's not get it from the list question from where you will get it again simple DB dot you can say uh get all this is the method I want so in the database I want to say get all but unfortunately we don't have a method so what you can do is you can say create method get all so we got the method and this should return a list so let me create a list here so I

06:09:24
will say list of products and we'll say products equals to new okay so people who are finding it difficult my suggestion would be First Watch the jdbc video and then then also watch the maven video so jdbc video and Maven video then watch the entire session you will love it you will see the power of Frameworks and how everything works okay so basically what we're doing is instead of getting data from a list last time we were getting data from the list right this time we want to fresh data from the database so

06:09:54
here we are saying we creating a list and let's fill this list from the values which is coming from database okay how do we do that so just say return product so that you will get rid of errors this time it will return empty let's let me try that let me see if I get the empty here let's run this and you can see we are not getting any data because it's empty the list is empty so how do we fill this it's very simple again use the same thing let's copy this because we are going to query the database paste

06:10:28
what are the changes the first change you're going to make is the query we're not going to insert this time we're going to select so we'll say select star uh okay we are not getting star we are going to mention the column name because we don't want ID there so we'll say name type place and warranty fromom table name is product and we don't have to apply well Clause is because we want all the data so basically we are fing everything from database right and and let's execute the query now since we

06:10:58
don't have any question mark here you can skip the entire section from here and simply say execute the difference is execute will not return the data we want data right so execute query is something you can use to to get the data and what it returns is the result set so it will return you a result set and what you can do is you can get data from this result set and store it in a list but the problem is it will give you uh d data in the result set format you have to convert that okay cool so basically we

06:11:31
want to F data from result set and set it in the products here because that's what we returning here right so how do you add so basically we can run a loop because if you say RS dot there's no method using which you can get all the data and save it in a list we have to do it manually one by one uh there's no option so what I will do is I will use a loop here and F data one by one so you can say fesh dat till the end and every time you fet data you can simply okay what we have to do is we have to create

06:12:02
a product object because okay I know we don't have time to explain this now but just go with the flow everyone so what we trying to do is we want to create a list of products right so we will be having multiple products so every time you run this Loop create a product object and it is highly recommended make sure that you watch the video of jdbc and Maven trust me this will be very very easy okay cool so uh let's get the product object and save each value in the product values okay uh so how do we

06:12:33
update the value so we have to say p do set name and this is coming from rs. getet string from the First Column so since we are getting multiple columns right this is the First Column second column third column fourth column then we can say p do Set uh type this is coming from second column so I will say get string column 2 p set place you know this is like a beautiful poem from a language which you don't understand right it sounds beautiful but you're not sure what is happening three then P do set warranty

06:13:11
and rs. get string or get integer this time it is integer and we are saying four okay so we were able to set the value for the product now once you get all the values just add the product in the product so we say products do add P now let's say you have have 15 values right so in the database now this while loop will run 15 times and every time you will get the new product object you are setting the values which is coming from database and adding to a list the same list goes back to the product

06:13:41
service the same list goes back to the main and in the main you are printing all the values let's see if this works run and voila can you see that this time you're getting data from the database is this making sense at least you are getting the gist of it right how things works out so yeah so basically we were able to fetch all the data right now same thing can be done for other things as well uh for the other methods which we have here so you can see we have it for uh get all we we can also do it for

06:14:10
the get product name for a particular name we just have to put say we have to change the query here okay cool so let's get back here and what we have to do is so yeah you can also do it for the uh get get a product with a name but that is your assignment okay so write SQL query and convert this into uh uh database I mean convert this so that you will get data from database make sense now I want to show you a real magic okay and that magic is how do we change this project into a spring Boot and a data jpa we'll compare this code

06:14:49
from the code which we'll write in the spring boot okay now I don't want to teach you entire spring boot now uh so yeah I don't want to teach you entire uh spring boot but I want to show you the

power of spring Boot and what it can do for you okay uh so how do you work with a spring boot project the first thing is if you want to create a project of spring boot you can go to inj and you can say file new when you create a project you have to make sure that you select spring boot but there should be

06:15:20
an option in your IDE I think my particular IDE is the ultimate version yeah so I have ultimate version here uh you will not get this options the spring initializer option I guess in the community version so if you want to do if you want to get this project even if you have a community version that's completely fine what you need to do is you need to go to the website and uh he where the other file went yeah so you have to go to the website called start. spring.io okay and here you have to select so you

06:16:03
have to select the project type first which is Maven in this case the language it supports is Java cotlin groovy we'll select Java here version whichever is default you can go for 3.1 and then you have to mention your group ID so that represents your company ID basically a unique one so this is com. telescope and then the project name so project name is product demo we'll say product spring you can change this as well uh make the packaging as jar and the version so make sure that whatever

06:16:38
version you have in your machine select that and then from here you have to add dependencies now what are the things we need so basically the first thing you need here is um what you need you need a post driver so post SQL driver added it next you need something called a data jpa because I wanted to show you the part of it yeah this is called a spring initialiser so add these two dependencies post G and spring jpa and click on generate that's it your job will be done you just have to unzip this

06:17:15
okay so click here unzip it I will just hide this so that I don't share everything uh it's in downloads okay nothing is nothing is risky there okay cool so you can see I have this file here what I will do now is I will go back to my inj I guess I already have the in DB here let's use this window I will open this project which I've downloaded which is there in downloads it's there in product spring and yeah open this project this window Okay cool so you can see we got our spring project here and trust me once

06:17:56
you see spring project you will say this is easy very very easy okay okay so if you go back to your pom.xml uh this is what you have you can see we got two dependencies here in fact we have three dependencies uh you got data jpa if you can see you got postgress and you got the testing part testing framework okay cool now in SRC if you go to main you have Java and this is your basically main application whatever code we have in main you will be writing that here expand this I will explain this in some

06:18:36
time okay this part so what I'm going to do is here I'm going to uh copy the code which we had here so let's you go back to main and let's not add everything because we already have that in database let's execute this part only this part we are fetching all the products how do I fetch all the products just paste it here the question is will this work uh of course not because we don't have a service class yet so how do we get the service class it's very simple you can just copy this service

class and

06:19:14
paste it here so you can see we got a service class which is here we also need a product class paste here okay so we got the product we got the service uh service says it is searching for product product DB okay we'll we'll get that and you know what I will do I will remove product DB and in get products let's say okay let's solve this problem later let's solve this errors later and just to keep it more simple I'll remove this part just to make it easy and yeah cool okay uh so the

06:19:57
problem is we don't have a product service class the object we don't have the object of product service so let's go back here and say product service let's create the object for it as service equal to new product service okay and then in this we have the object of product we don't need a list as at this point so we can remove the list as well and okay now one thing to remember about spring framework now spring framework provides you one of the most important feature in programming which

06:20:26
is the springing the dependency injection now what is dependency injection if you go back to the main method you can see we are creating an object of product service when you say new keyword and when you mention the Constructor basically you as a programmer you are creating an object by yourself is it a bad thing yes is because when you create the object it is your responsibility to destroy it basically you have to manage the life cycle of an object spring framework says you don't have to create the object I

06:21:00
will I will do it for you trust me Spring framework says you don't have to create the object I will do it for you let me show you what I'm saying what I will do is I will just comment all the methods here comment and let's create a very simple method so I will say public void show and in this show basically I'm saying in show okay and we all all know that if you want to create if you want to call this show you need to First create the object let me come in this section as well yeah so first you need

06:21:35
to create object of product service then only you can call it right example I can just go back here and I can say service do show this will work let me try run and you can see oh it is giving you errors because we have added this dependency of database okay okay okay let me comment this post and data jpa for some time only data jpa commenting should work when I comment it and if I go back to my application and okay what did I just comment it oh yeah if I comment that then we have to get this spring boot dependency

06:22:22
okay you know what I will do just trust trust me just trust me because I have to make a lot of changes if you if I want to achieve that because I have to get this spring boot dependency then if I commented uh let just reload the MAV dependencies okay so the thing is you have to say service do so that it will work right but then spring boot says you don't have to create the object you can just remove this and it will still work how step one you have to tell spring framework to create the object for you

06:22:53
spring framework will give you the object and for that just on top of your class say add component

so what you're doing is you're telling your spring framework to create the object so spring will handle it for you you don't have to handle it uh ignore the XML file at this point if you're finding it confusing and make sure that you do it by yourself then it will make much more sense okay cool uh and then go back to your see the problem is we are doing so many things at the same time in less amount of time

06:23:20
it it does create some confusion okay so what I will do here is I will just try to okay so how do I get this object from spring so what you do is when you talk about the first line spring provides you something called Spring container so every time you run this line I mean the moment you run this line it will create a container for you imagine a container as a box where the objects are already created so you don't have to create the object objects are created and if you want to learn everything in detail uh

06:23:48
you will find the link in description about this spring video just watch the video you will get the complete idea now this basically gives you a application context object so let's use it so this run returns you the object of application context let's just use it and using that context now since the object is created right so the object of product service is already created by Spring framework and it is available for to use how do you get the object it's very simple you simply say context Dot

06:24:17
and you say spring framework give me the object okay so spring framework get bean and mention which class object you want I want the object of product service service do class so you mention which class object you want and spring will give you the object yeah this is called dependency injection and now this should work but the problem is I can't run it because you know there is there are certain properties you have to mention so I will not run it now so get bin will give you the object but this will work

06:24:47
trust me okay so what we say what we're saying is we want to get the data right so from the service uh let me just uncommon this part so from the service we are saying get all products which is here okay we are not going to go do add product right which is already there data is already there let's use get all products uh okay so you can see we are saying get all products and this will work only when your application is connected to a database and that to with the help of queries okay and you can see we don't

06:13:41
service the same list goes back to the main and in the main you are printing all the values let's see if this works run and voila can you see that this time you're getting data from the database is this making sense at least you are getting the gist of it right how things works out so yeah so basically we were able to fetch all the data right now same thing can be done for other things as well uh for the other methods which we have here so you can see we have it for uh get all we we can also do it for

06:14:10
the get product name for a particular name we just have to put say we have to change the query here okay cool so let's get back here and what we have to do is so yeah you can also do it for the uh get get a product with a name but that is your assignment okay so write SQL query and convert this into uh uh database I mean convert this so that you will get data from database make sense now I want to show you a real magic okay and that magic is how do we change this project into a

spring Boot and a data jpa we'll compare this code

06:14:49
from the code which we'll write in the spring boot okay now I don't want to teach you entire spring boot now uh so yeah I don't want to teach you entire uh spring boot but I want to show you the power of spring Boot and what it can do for you okay uh so how do you work with a spring boot project the first thing is if you want to create a project of spring boot you can go to inj and you can say file new when you create a project you have to make sure that you select spring boot but there should be

06:15:20
an option in your IDE I think my particular IDE is the ultimate version yeah so I have ultimate version here uh you will not get this options the spring initializer option I guess in the community version so if you want to do if you want to get this project even if you have a community version that's completely fine what you need to do is you need to go to the website and uh he where the other file went yeah so you have to go to the website called start. spring.io okay and here you have to select so you

06:16:03
have to select the project type first which is Maven in this case the language it supports is Java cotlin groovy we'll select Java here version whichever is default you can go for 3.1 and then you have to mention your group ID so that represents your company ID basically a unique one so this is com. telescope and then the project name so project name is product demo we'll say product spring you can change this as well uh make the packaging as jar and the version so make sure that whatever

06:16:38
version you have in your machine select that and then from here you have to add dependencies now what are the things we need so basically the first thing you need here is um what you need you need a post driver so post SQL driver added it next you need something called a data jpa because I wanted to show you the part of it yeah this is called a spring initialiser so add these two dependencies post G and spring jpa and click on generate that's it your job will be done you just have to unzip this

06:17:15
okay so click here unzip it I will just hide this so that I don't share everything uh it's in downloads okay nothing is nothing is risky there okay cool so you can see I have this file here what I will do now is I will go back to my inj I guess I already have the in DB here let's use this window I will open this project which I've downloaded which is there in downloads it's there in product spring and yeah open this project this window Okay cool so you can see we got our spring project here and trust me once

06:17:56
you see spring project you will say this is easy very very easy okay okay so if you go back to your pom.xml uh this is what you have you can see we got two dependencies here in fact we have three dependencies uh you got data jpa if you can see you got postgress and you got the testing part testing framework okay cool now in SRC if you go to main you have Java and this is your basically main application whatever code we have in main you will be writing that here expand this I will explain this in some

06:18:36

time okay this part so what I'm going to do is here I'm going to uh copy the code which we had here so let's you go back to main and let's not add everything because we already have that in database let's execute this part only this part we are fetching all the products how do I fetch all the products just paste it here the question is will this work uh of course not because we don't have a service class yet so how do we get the service class it's very simple you can just copy this service class and

06:19:14
paste it here so you can see we got a service class which is here we also need a product class paste here okay so we got the product we got the service uh service says it is searching for product product DB okay we'll we'll get that and you know what I will do I will remove product DB and in get products let's say okay let's solve this problem later let's solve this errors later and just to keep it more simple I'll remove this part just to make it easy and yeah cool okay uh so the

06:19:57
problem is we don't have a product service class the object we don't have the object of product service so let's go back here and say product service let's create the object for it as service equal to new product service okay and then in this we have the object of product we don't need a list as at this point so we can remove the list as well and okay now one thing to remember about spring framework now spring framework provides you one of the most important feature in programming which

06:20:26
is the springing the dependency injection now what is dependency injection if you go back to the main method you can see we are creating an object of product service when you say new keyword and when you mention the Constructor basically you as a programmer you are creating an object by yourself is it a bad thing yes is because when you create the object it is your responsibility to destroy it basically you have to manage the life cycle of an object spring framework says you don't have to create the object I

06:21:00
will I will do it for you trust me Spring framework says you don't have to create the object I will do it for you let me show you what I'm saying what I will do is I will just comment all the methods here comment and let's create a very simple method so I will say public void show and in this show basically I'm saying in show okay and we all all know that if you want to create if you want to call this show you need to First create the object let me come in this section as well yeah so first you need

06:21:35
to create object of product service then only you can call it right example I can just go back here and I can say service do show this will work let me try run and you can see oh it is giving you errors because we have added this dependency of database okay okay okay let me comment this post and data jpa for some time only data jpa commenting should work when I comment it and if I go back to my application and okay what did I just comment it oh yeah if I comment that then we have to get this spring boot dependency

06:22:22
okay you know what I will do just trust trust me just trust me because I have to make a lot of changes if you if I want to achieve that because I have to get this spring boot dependency then if I commented uh let just reload the MAV dependencies okay so the thing is you have to say service do so that it will work right but then spring boot says you don't have to create the object you can

just remove this and it will still work how step one you have to tell spring framework to create the object for you

06:22:53
spring framework will give you the object and for that just on top of your class say add component so what you're doing is you're telling your spring framework to create the object so spring will handle it for you you don't have to handle it uh ignore the XML file at this point if you're finding it confusing and make sure that you do it by yourself then it will make much more sense okay cool uh and then go back to your see the problem is we are doing so many things at the same time in less amount of time

06:23:20
it it does create some confusion okay so what I will do here is I will just try to okay so how do I get this object from spring so what you do is when you talk about the first line spring provides you something called Spring container so every time you run this line I mean the moment you run this line it will create a container for you imagine a container as a box where the objects are already created so you don't have to create the object objects are created and if you want to learn everything in detail uh....give important and teach me like this only in very very simple

Samiksha... 🤍
I will now explain **ONLY the important part**, in **very very simple language**, just like the teaching style you want.

You pasted a *very big transcript*, but the important concept you want here is:

⭐ "How to connect Java to a database (PostgreSQL) using JDBC — step by step, extremely simple."

⭐ "Why we create ProductDB class?"

⭐ "How save() and getAll() work?"

⭐ "How Spring Boot simplifies everything later?"

So I will explain EXACTLY these parts in super simple steps.

🌼 PART 1 — Why create ProductDB class?

Imagine:

- **ProductService** = Logic (filters, search, business rules)
- **ProductDB** = Only database work

You never mix kitchen + bedroom.
Similarly we don't mix business logic + database logic.

So ProductService says:

👉 *“Hey ProductDB... save this product in DB, I don’t care how.”*

This is clean architecture.

🌼 PART 2 — ProductService calls ProductDB

```java
ProductDB db = new ProductDB();
db.save(p);
```

Meaning:

➡️ ProductService gives the product
➡️ ProductDB puts it inside the database

Very simple.

🌼 PART 3 — In ProductDB, we create a Connection

To talk to a database, Java needs a telephone line.
That telephone line = **Connection object**

```java
Connection con = null;
```

But we need to fill it.

🌼 PART 4 — How to open connection (VERY SIMPLE)

This happens in ProductDB constructor:

```java
public ProductDB() {
    try {
        con = DriverManager.getConnection(
            "jdbc:postgresql://localhost:5432/telescope",
            "postgres",
            "0000"   // whatever your password is
        );
    } catch(Exception e) {
        e.printStackTrace();
    }
}
```

Explanation in simple way:

- **`jdbc:postgresql://`** → tell Java the type of database
- **`localhost`** → database is on your computer
- **`5432`** → PostgreSQL default port

- **`telescope`** → your database name
- **`"postgres"`** → username
- **`"0000"`** → password

Connection success = Java has reached database.

🌼 PART 5 — How save() method works (in simple words)

Goal → Insert product into database table.

Your table has:

- name
- type
- place
- warranty

So SQL is:

```
pgsql

insert into product (name, type, place, warranty)
values (?, ?, ?, ?)
```

Why **?**?
Because we will fill them later from Java.

Actual Java code:

```
java

String query = "insert into product (name, type, place, warranty) values (?, ?, ?, ?)";

PreparedStatement st = con.prepareStatement(query);

st.setString(1, p.getName());
st.setString(2, p.getType());
st.setString(3, p.getPlace());
st.setInt(4, p.getWarranty());

st.execute();
```

Very simple meaning:

- Prepare the query
- Replace ? with real data
- Execute it

Database stores the product.

🌼 PART 6 — How getAll() works (super simple)

Goal → Fetch all rows from database.

SQL:

```pgsql
select name, type, place, warranty from product
```

Java:

```java
String query = "select name, type, place, warranty from product";

PreparedStatement st = con.prepareStatement(query);
ResultSet rs = st.executeQuery();
```

Now `rs` contains ALL rows.

We read them one-by-one:

```java
while (rs.next()) {
    Product p = new Product();
    p.setName(rs.getString(1));
    p.setType(rs.getString(2));
    p.setPlace(rs.getString(3));
    p.setWarranty(rs.getInt(4));

    products.add(p);
}
```

This fills your list.

Then we return it.

🌼 PART 7 — VERY SIMPLE FLOW (beginner-friendly)

When you run Main:

1️⃣ Main calls ProductService
2️⃣ ProductService calls ProductDB
3️⃣ ProductDB connects to database
4️⃣ save() → inserts data
5️⃣ getAll() → fetches data
6️⃣ Service returns data
7️⃣ Main prints data

That’s it.
You are now doing REAL backend development.

🌼 PART 8 — Why Spring Boot is MAGIC (Explained like for child)

Normal JDBC:

- You write connection code
- You write SQL
- You write loops
- You map ResultSet → Objects manually
- You handle errors
- You handle resources

Spring Boot + JPA:

- You remove entire ProductDB class
- You remove all SQL
- You remove connection code
- You remove loops
- You just write:

```java
productRepo.save(product);
productRepo.findAll();
productRepo.findByName("Laptop");
```

Spring Boot internally:

✔️ Creates connection
✔️ Creates SQL
✔️ Executes SQL
✔️ Converts rows → objects
✔️ Handles exceptions

You write ZERO code.

🌼 PART 9 — Do you want the next part?

I can continue in **your preferred very very simple style** and explain:

1️⃣ Turning this whole project into SPRING BOOT + JPA (easy magic)

2️⃣ How @Entity, @Id, @GeneratedValue work

3️⃣ How to create Repository interface

4️⃣ How Spring automatically fetches data

5️⃣ How to remove ProductDB completely

6️⃣ How to run SQL inside Spring Boot

7️⃣ How to create REST API for this project

Just tell me:

👉 **“Explain Spring Boot version also in same simple way.”**

or

👉 **“Continue next part.”**

I will continue exactly in the same simple, clear, baby-level teaching style.

06:23:48
you will find the link in description about this spring video just watch the video you will get the complete idea now this basically gives you a application context object so let's use it so this run returns you the object of application context let's just use it and using that context now since the object is created right so the object of product service is already created by Spring framework and it is available for to use how do you get the object it's very simple you simply say context Dot

06:24:17
and you say spring framework give me the object okay so spring framework get bean and mention which class object you want I want the object of product service service do class so you mention which class object you want and spring will give you the object yeah this is called dependency injection and now this should work but the problem is I can't run it because you know there is there are certain properties you have to mention so I will not run it now so get bin will give you the object but this will work06:24:47
trust me okay so what we say what we're saying is we want to get the data right so from the service uh let me just uncommon this part so from the service we are saying get all products which is here okay we are not going to go do add product right which is already there data is already there let's use get all products uh okay so you can see we are saying get all products and this will work only when your application is connected to a database and that to with the help of queries okay and you can see we don't

06:25:22
have any file here which is which has a datab connection but let me create a file here because we want it DB right so what I will do is I will just right click here and say create a class and we'll name this as product maybe you can say uh DB or whatever name you you use doesn't matter and in this class you have to mention all the queries but then if you really want to use queries here then why you are using data JP and all these features let me show you something let's not even create a class let's

06:25:54
create the interface okay okay and the only thing you will do here is you will extend something called jpa repository now jpa repository is a interface which will help you with the concept of omm now on this channel there's a there's a video on om please watch it hush is not here so you can't add the add that in description but you can search on what is or and watch the video you will get much more idea but ultimately when you see the output for this you will get the the gist of it now

repository says tell

06:26:27
me which class you're working with so I'm working with product and what is the type of your primary key the primary key is integer so we simply create this particular uh interface here and no configuration okay we we have not done any code here but see the magic and okay now this is the exciting Point people who are still watching just see the magic if I go back to product service and let's create the reference of product DB so DB DB just a reference and DB dot can you see that we got methods

06:27:02
just say find all no need for any SQL query no need to do everything which we have done here if I show you the code what we have done in the earlyer project is this if you want to get all you have to line you have to write this number of lines with this no line this is what data jpa is doing for you it's not like behind the scenes there's no SQL query because the only way you can interact with your database is with the help of SQL but this is happening automatically is it making sense okay but the thing is

06:27:35
we got the method but unfortunately your your your application has no idea about the configuration nothing because nowhere we are mentioned which dbms you're working with is it postgress is it Mas or is it Oracle we have not even mentioned database name we have not even mentioned username password we have to configure those somewhere now where is that somewhere if you can see there's a resource folder here in the application. properties this is where you mention all the properties now what's the property

06:28:05
I'm talking about I'm talking about the URL the URL is for the spring data source. Ur and the UR is same which we have written before which is DBC colon postgress let me just copy it because this is the URL which we want copy and paste it here next thing you have to mention is the username so spring data source username which is pis single s and we also need a password so spring data source password which is 0000 and that's it once you mention this let's see if this works run oh we got errors I thought it

06:28:57
will work okay it says there are certain things which are missing what is that missing thing it says error creating a beIN with product DB oh okay the thing is in the product service we are creating a reference for product DB right where is the object where is the object that's the problem so thing is as we mentioned for product service remember we have mentioned component here in the same way in the product DB as well on top of it we have to say this is a repository because we are working with repository here and in product

06:29:27
service if if I want see the the thing is just by mentioning this spring will not know that you want this object you have to say hey Spring I want an object of product DB and for that you have to say Auto wire search it and give me the object okay if you want to know more about aut Wire Watch the video which is there in the description about spring framework okay if you want to know how I got this full thing when I just mention user just type user it will give you prompt and it happens with some of the

06:29:54

IDS okay and it should work now let's rerun the code okay now we have different error and I love errors see the thing is it says and sometime you learn byas right it says everything is good now I got the object but the problem is this product class is not managed it is not it's still not a be and also there's one more thing which you have to remember when you talk about the product it is not just a simple class right this is trying to map so basically your table and your class is getting mapped and to achieve that you

06:30:25
have to say add entity here next you also need to specify the primary key which is the primary key here which is private in ID that's your primary key and the the do to mention that it's a primary key you have to say at ID is it ID yeah so at ID and then we have to also mention that this value is actually autogenerated okay so how do you get this autogenerated uh so ignore this part because this is just something which you can see later you have to say autogenerated uh generated value equal

06:30:55
to uh you have to set a strategy if I'm not wrong you have you have to set the strategy and the strategy is generated type uh identity sequin should also work I guess but let's say identity and also we need to get the data Setters for this for ID right click generate get a set for ID you know we are doing so much and so much of thing in one video and you need to rewatch this things are looking good now let's see if this works run and what VOA can you see that you got all the data from the database without writing

06:31:29
any SQL query yeah we can also add ID if you want to get the ID but this this should work now what are the things you can do you can also add you can also add the product because if you see if you go back to your product service and let's try to find what are the objects what are the methods we have so DB DB dot you can see we have find all we also have save we also got uh count how many column how many rows you have you can also delete if you want you can also say find one we have so many options there

06:31:59
is it making sense and just imagine just by using this your productivity will go on the next level and that's why in the industry spring framework is so famous so let me explain the entire flow once again if you want to create this project the first thing you do is you go to Spring uh start. spring.io mention all these properties here which is uh mavin Java whatever springboard version you want to use make sure that you don't use Snapshot because this creates some issues uh and then mention the group ID

06:32:33
mention the project name and add these two dependencies the post quiz SQL driver and spring data jpa next you need to uh copy the files which you have already created Service uh let's start with the XML file first in the XML file make sure that you have these two dependencies or three dependencies at least these two is these two are very important for us oh I got tired in not only next in the uh product service you have to make sure that this is component because I want spring to uh give the

06:33:02
project yeah sorry uh I want spring framework to give me the object of product service and uh I can use something called Product DB which is a interface not a class and it is extending with jpa repository and with this you can perform the operations the cut operations with the uh with this

particular interface you can fresh data you can save data you can you can update data you can delete data and we are able to do that but then you have to also set some properties which is application. properties which

06:33:31
we have done here so go with the video once again uh I will try to share this code today itself hey yeah you can also use service here for different layers basically we use different annotations doesn't matter it works so all the uh entity class or normal classes you can have component entity class will have entity and uh service class can have service so got the idea what is spring data jpa so all these operations where you don't have to add SQL query is happening with the help of spring data

06:34:00
jpa so before spring we used to do that with the help of hibernate and if you want to learn about hibernate there's a video on hibernate as well and know if you check out the channel there are so many videos in last 5 to six years sometime even I forgot what videos I have your code will be available so after after some time check out the description for this particular live session uh you will find the code repo as a title and then there will be a GitHub link and now today we are going to build the same project which we have

06:34:27
done before so that we can it it will make much more sense okay what we are doing but this time we are going to use uh MVC okay so basically we are converting our project into a web project okay that will be cool right so let's get started and let me show you the code what we have done till now and let's talk about the code and then we'll also see what we're going to do today in this particular uh in the image okay so if you if I want to go through one by one uh basically Bally we have this

06:34:55
number of classes here let me just move it here yeah uh and also move it this side so basically we we have a product class it's all about this class which is product class and in this we have used four variables basically and for those four variables of course we need to get gets and Setters in fact um before this session around 600 p.m. or 5:00 p.m. IST I have uploaded a video a shot on YouTube which talks about uh the longbo how do you reduce this code okay so if you have not checked it out do check it out later so we'll we'll

06:35:31
also convert this into a longb but we'll see later how do we do that uh so this is the class which you want to work with so every product in my office basically it's a that's a product right even if you talk about this remote that is a product now what I want to achieve with this is with this product basically I want to store I want to have a place where I can check where a product is actually is example there are some products on this St C in fact lot of products because I'm going live I have a

06:25:22
have any file here which is which has a datab connection but let me create a file here because we want it DB right so what I will do is I will just right click here and say create a class and we'll name this as product maybe you can say uh DB or whatever name you you use doesn't matter and in this class you have to mention all the queries but then if you really want to use queries here then why you are using data JP and all these features let me show you something let's not even create a class let's

06:25:54

create the interface okay okay and the only thing you will do here is you will extend something called jpa repository now jpa repository is a interface which will help you with the concept of omm now on this channel there's a there's a video on om please watch it hush is not here so you can't add the add that in description but you can search on what is or and watch the video you will get much more idea but ultimately when you see the output for this you will get the the gist of it now repository says tell

06:26:27
me which class you're working with so I'm working with product and what is the type of your primary key the primary key is integer so we simply create this particular uh interface here and no configuration okay we we have not done any code here but see the magic and okay now this is the exciting Point people who are still watching just see the magic if I go back to product service and let's create the reference of product DB so DB DB just a reference and DB dot can you see that we got methods

06:27:02
just say find all no need for any SQL query no need to do everything which we have done here if I show you the code what we have done in the earlyer project is this if you want to get all you have to line you have to write this number of lines with this no line this is what data jpa is doing for you it's not like behind the scenes there's no SQL query because the only way you can interact with your database is with the help of SQL but this is happening automatically is it making sense okay but the thing is

06:27:35
we got the method but unfortunately your your your application has no idea about the configuration nothing because nowhere we are mentioned which dbms you're working with is it postgress is it Mas or is it Oracle we have not even mentioned database name we have not even mentioned username password we have to configure those somewhere now where is that somewhere if you can see there's a resource folder here in the application. properties this is where you mention all the properties now what's the property

06:28:05
I'm talking about I'm talking about the URL the URL is for the spring data source. Ur and the UR is same which we have written before which is DBC colon postgress let me just copy it because this is the URL which we want copy and paste it here next thing you have to mention is the username so spring data source username which is pis single s and we also need a password so spring data source password which is 0000 and that's it once you mention this let's see if this works run oh we got errors I thought it

06:28:57
will work okay it says there are certain things which are missing what is that missing thing it says error creating a beIN with product DB oh okay the thing is in the product service we are creating a reference for product DB right where is the object where is the object that's the problem so thing is as we mentioned for product service remember we have mentioned component here in the same way in the product DB as well on top of it we have to say this is a repository because we are working with repository here and in product

06:29:27
service if if I want see the the thing is just by mentioning this spring will not know that you want this object you have to say hey Spring I want an object of product DB and for that you have to say Auto wire search it and give me the object okay if you want to know more about aut Wire Watch

the video which is there in the description about spring framework okay if you want to know how I got this full thing when I just mention user just type user it will give you prompt and it happens with some of the

06:29:54
IDS okay and it should work now let's rerun the code okay now we have different error and I love errors see the thing is it says and sometime you learn byas right it says everything is good now I got the object but the problem is this product class is not managed it is not it's still not a be and also there's one more thing which you have to remember when you talk about the product it is not just a simple class right this is trying to map so basically your table and your class is getting mapped and to achieve that you

06:30:25
have to say add entity here next you also need to specify the primary key which is the primary key here which is private in ID that's your primary key and the the do to mention that it's a primary key you have to say at ID is it ID yeah so at ID and then we have to also mention that this value is actually autogenerated okay so how do you get this autogenerated uh so ignore this part because this is just something which you can see later you have to say autogenerated uh generated value equal

06:30:55
to uh you have to set a strategy if I'm not wrong you have you have to set the strategy and the strategy is generated type uh identity sequin should also work I guess but let's say identity and also we need to get the data Setters for this for ID right click generate get a set for ID you know we are doing so much and so much of thing in one video and you need to rewatch this things are looking good now let's see if this works run and what VOA can you see that you got all the data from the database without writing

06:31:29
any SQL query yeah we can also add ID if you want to get the ID but this this should work now what are the things you can do you can also add you can also add the product because if you see if you go back to your product service and let's try to find what are the objects what are the methods we have so DB DB dot you can see we have find all we also have save we also got uh count how many column how many rows you have you can also delete if you want you can also say find one we have so many options there

06:31:59
is it making sense and just imagine just by using this your productivity will go on the next level and that's why in the industry spring framework is so famous so let me explain the entire flow once again if you want to create this project the first thing you do is you go to Spring uh start. spring.io mention all these properties here which is uh mavin Java whatever springboard version you want to use make sure that you don't use Snapshot because this creates some issues uh and then mention the group ID

06:32:33
mention the project name and add these two dependencies the post quiz SQL driver and spring data jpa next you need to uh copy the files which you have already created Service uh let's start with the XML file first in the XML file make sure that you have these two dependencies or three dependencies at least these two is these two are very important for us oh I got tired in not only next in the uh product service you have to make sure that this is component because I want spring to uh give the

06:33:02
project yeah sorry uh I want spring framework to give me the object of product service and uh I can use something called Product DB which is a interface not a class and it is extending with jpa repository and with this you can perform the operations the cut operations with the uh with this particular interface you can fresh data you can save data you can you can update data you can delete data and we are able to do that but then you have to also set some properties which is application. properties which

06:33:31
we have done here so go with the video once again uh I will try to share this code today itself hey yeah you can also use service here for different layers basically we use different annotations doesn't matter it works so all the uh entity class or normal classes you can have component entity class will have entity and uh service class can have service so got the idea what is spring data jpa so all these operations where you don't have to add SQL query is happening with the help of spring data

06:34:00
jpa so before spring we used to do that with the help of hibernate and if you want to learn about hibernate there's a video on hibernate as well and know if you check out the channel there are so many videos in last 5 to six years sometime even I forgot what videos I have your code will be available so after after some time check out the description for this particular live session uh you will find the code repo as a title and then there will be a GitHub link and now today we are going to build the same project which we have

06:34:27
done before so that we can it it will make much more sense okay what we are doing but this time we are going to use uh MVC okay so basically we are converting our project into a web project okay that will be cool right so let's get started and let me show you the code what we have done till now and let's talk about the code and then we'll also see what we're going to do today in this particular uh in the image okay so if you if I want to go through one by one uh basically Bally we have this

06:34:55
number of classes here let me just move it here yeah uh and also move it this side so basically we we have a product class it's all about this class which is product class and in this we have used four variables basically and for those four variables of course we need to get gets and Setters in fact um before this session around 600 p.m. or 5:00 p.m. IST I have uploaded a video a shot on YouTube which talks about uh the longbo how do you reduce this code okay so if you have not checked it out do check it out later so we'll we'll

06:35:31
also convert this into a longb but we'll see later how do we do that uh so this is the class which you want to work with so every product in my office basically it's a that's a product right even if you talk about this remote that is a product now what I want to achieve with this is with this product basically I want to store I want to have a place where I can check where a product is actually is example there are some products on this St C in fact lot of products because I'm going live I have a

06:35:58
stream deck for the switching between uh the screens I have a Mac Studio for the streaming I have my personal laptop here uh to check everything is going well I have iPad where I write something and I don't know why this light is sitting here I also have a light here that's huge okay so it gives

you a good light on the face uh so basically uh we have so many items here right so sometime we keep this item somewhere and we we lose it right so we have to search where that particular item is and for that we have

06:36:29
this uh database or this application which stores where it is and that's why you can see we have a name typ place and warranty because I also want to know if the current product is still in warranty or not now to manage it uh what if you want to add a product what if you want to remove a product what if you are not remove we have not done that what if you want to add a product what if you want to search something what the we the product particular product is and also I want to know uh in a particular place

06:36:59
how many products I have what are the products we have and if you want to put that logic we can do that in here so we have this product service class and the reason I have created different classes here of course you can write everything in one class because that's what we do uh when we start learning coding right uh in fact I remember when I used to start when I started uh my programming journey I used to write everything in uh in one particular file it's good it works the problem is if you want to

06:37:28
change something and you're not sure what you're changing right and that's why creating different layers makes much more sense so what layers we have so basically this is my main do Java this is where I execute something so if you want to call something and if you want to do all the execution stuff which calling methods and stuff you will do that in the main class of course you can name this class anything you want but since this is a console based application which we already built we

06:37:57
have this main method here and in this basically we are getting all the products now how we are getting the products that's not the concern of meain it's just it just says I want the product give me the products uh same goes for other methods as well if you want to search a particular product uh if you want to search for a place if you want to add the products so basically we have the we have the calling here the actual logic goes here so you can see we have a product service here this is

06:38:22
where the actual l logic goes okay but now since we want to store this data in the database right see not every application with have a database and if you're using a database of course you can write your database connectivity code so doesn't matter which language you work with which database you work with we have to connect to a database right we have to follow certain certain steps in the same way here if you want to write your database connections don't do it in service class service class

06:38:50
simply says this is where your logic will be if you want to process something uh example let's say if you want to add two numbers right so from a main method you will say hey add method take this two values which is two and five just add it the actual calculation will be happening in the service so all the processing will be done in the service but what if you want to get something from database and that's why we have one more layer here now this particular layer actually communicates with your

06:39:17

database right uh in fact in the earlier session we were able to implement uh okay I think I have done different thing yeah this was the project right because we have also implemented spring let's talk about this one uh but things remain same right if you if you go back to your uh previous Cod this is day two which I'm talking about U so you can see we are writing all the SQL queries now what we have done in the previous session so basically we are doing a quick revision of what we have done uh but if you if

06:39:46
you go back to our previous session we tried to reduce the code right we tried to say let's use certain tools and some Frameworks so that we we don't have to do everything by ourself now my question to you is is it a good idea to use Frameworks to reduce the code what do you think you know some of the people I know they say Hey you know uh we as a programmer we should write everything by ourself right what do you think that's a good idea whatever the application does don't Tak help don't take help from

06:40:14
anything else just build it by yourself okay let me talk about the first problem which is if you try to do everything by yourself there are a lot of things you have to do by yourself building your own OS because ultimately your application runs on the OS uh building your own runtime that means your jvm and J uh building uh I mean not using any Library example here for database connection as well we have imported some packages don't do that do it by everything by yourself right and we can't do that

06:40:40
right we we want to complete the project in a month or two or maybe in a year if it just complex project if you try to do everything by yourself it will take months even to write a simple database connection the second reason is uh before I talk about the second reason I have a question for you and just complete the sentence I have used this multiple times so even if you know it you can answer that or if you don't know about it think about it it's a funny thing not a uh serious issue or serious

06:41:09
thing so my question is if debugging is removing bugs okay that's what we say right debugging is removing bugs then what is coding if debugging is removing bugs then coding is adding bugs so what I'm saying is every time time you write one line of code that is a potential bug let me show you this line can be a potential bug this line can be a potential bug it will not it may not come up when you run the application but may based on the different uh input from the user it might behave differently and

06:41:42
that's why we do testing right so every time you write your application it's important for you to test your application that's very important so one way to reduce the testing and uh one way to reduce the potential works is don't write the code write minimum lines of code what if the spr the framework which you working with says hey you know you don't have to write all these lines I will give it to you behind the scene the good thing is all this particular uh Frameworks they are well tested and they

06:42:13
are getting used in the industry from a long time so at least you can trust them that they are not creating issues okay I'm not saying it is 100% safe that there have been some ISS some uh cases where you have used certain libraries and they found some issues with the library and because of

which the Hat attack happened uh you might have heard about log 4J in fact in the spring framework also there was some I forgot the library but if anyone knows let me know in the chat uh it was having some

06:42:41
issues and again it's not like the issue stays it gets rectified but uh it happens I'm not saying it is 100% safe but try to write less amount of code okay cool uh so this is what we have done in in the database but then in the yesterday session we have seen something called a spring framework Now using spring framework spring provides you something called a dependency injection which means you don't have to create the object you simply have to use the object uh example if I want the object of

06:43:12
product service in the previous example uh on day two basically we were able to create the object with the help of new keyword but the moment you use a new keyword that means you are handling the object object you creating object you are doing everything with it so in in uh spring what we doing is we are not creating object by yourself we are simply saying there's a spring container so spring Frameworks actually create a container for you imagine a container as a box which has lot of different objects

06:43:39
and you simply Say Hey I want that object example let's say if you want a phone of course you will not manufacture the phone by yourself you will simply go to a shop and buy the phone in the same way spring conna says I have all the objects which you need just ask for it it okay and that's what we doing we saying hey Spring get me a bean get me the object basically the objects in Spring containers are called Bean of course it has more logic to it but let's not talk about the logic at this point

06:44:08
imagine being as an object okay and then we are getting the object of this particular class which is uh product service and then you got the object and you can use the object for your purpose right and we able to call the methods now the beauty is there's nothing Chang change in the service right you can see we have all the same methods of we have commented it but it should it should work and uh product service needs the object of product DB right of course if you have a class which is calling

06:44:37
another class methods you need object for it and you don't have to say new keyword what you can simply do is all the classes or all the interfaces which is repository basically uh what you can do is you can write a annotation now let's not discuss the difference between repository service component again it will take another two hours just to understand it but you know just a gist basically spring will say okay the moment you mention repository the moment you mention uh service the moment you

06:45:10
mentioned component where we have used it uh entity and you can also use component what spring will do is every class which has this annotation on top of it it will create object for you and you you don't have to get object by yourself spring will give it to you and you simply say give me the object and one way to do that is by using autowired now how exactly it is searches let's not talk about those things okay but then it will it will give it to you now what we are going to do today is we will be

06:35:58
stream deck for the switching between uh the screens I have a Mac Studio for the streaming I have my personal laptop here uh to check everything is going well I have iPad where I write something and I don't know why this light is sitting here I also have a light here that's huge okay so it gives you a good light on the face uh so basically uh we have so many items here right so sometime we keep this item somewhere and we we lose it right so we have to search where that particular item is and for that we have

06:36:29
this uh database or this application which stores where it is and that's why you can see we have a name typ place and warranty because I also want to know if the current product is still in warranty or not now to manage it uh what if you want to add a product what if you want to remove a product what if you are not remove we have not done that what if you want to add a product what if you want to search something what the we the product particular product is and also I want to know uh in a particular place

06:36:59
how many products I have what are the products we have and if you want to put that logic we can do that in here so we have this product service class and the reason I have created different classes here of course you can write everything in one class because that's what we do uh when we start learning coding right uh in fact I remember when I used to start when I started uh my programming journey I used to write everything in uh in one particular file it's good it works the problem is if you want to

06:37:28
change something and you're not sure what you're changing right and that's why creating different layers makes much more sense so what layers we have so basically this is my main do Java this is where I execute something so if you want to call something and if you want to do all the execution stuff which calling methods and stuff you will do that in the main class of course you can name this class anything you want but since this is a console based application which we already built we

06:37:57
have this main method here and in this basically we are getting all the products now how we are getting the products that's not the concern of meain it's just it just says I want the product give me the products uh same goes for other methods as well if you want to search a particular product uh if you want to search for a place if you want to add the products so basically we have the we have the calling here the actual logic goes here so you can see we have a product service here this is

06:38:22
where the actual l logic goes okay but now since we want to store this data in the database right see not every application with have a database and if you're using a database of course you can write your database connectivity code so doesn't matter which language you work with which database you work with we have to connect to a database right we have to follow certain certain steps in the same way here if you want to write your database connections don't do it in service class service class

06:38:50
simply says this is where your logic will be if you want to process something uh example let's say if you want to add two numbers right so from a main method you will say hey add method take this two values which is two and five just add it the actual calculation will be happening in the service so

all the processing will be done in the service but what if you want to get something from database and that's why we have one more layer here now this particular layer actually communicates with your

06:39:17
database right uh in fact in the earlier session we were able to implement uh okay I think I have done different thing yeah this was the project right because we have also implemented spring let's talk about this one uh but things remain same right if you if you go back to your uh previous Cod this is day two which I'm talking about U so you can see we are writing all the SQL queries now what we have done in the previous session so basically we are doing a quick revision of what we have done uh but if you if

06:39:46
you go back to our previous session we tried to reduce the code right we tried to say let's use certain tools and some Frameworks so that we we don't have to do everything by ourself now my question to you is is it a good idea to use Frameworks to reduce the code what do you think you know some of the people I know they say Hey you know uh we as a programmer we should write everything by ourself right what do you think that's a good idea whatever the application does don't Tak help don't take help from

06:40:14
anything else just build it by yourself okay let me talk about the first problem which is if you try to do everything by yourself there are a lot of things you have to do by yourself building your own OS because ultimately your application runs on the OS uh building your own runtime that means your jvm and J uh building uh I mean not using any Library example here for database connection as well we have imported some packages don't do that do it by everything by yourself right and we can't do that

06:40:40
right we we want to complete the project in a month or two or maybe in a year if it just complex project if you try to do everything by yourself it will take months even to write a simple database connection the second reason is uh before I talk about the second reason I have a question for you and just complete the sentence I have used this multiple times so even if you know it you can answer that or if you don't know about it think about it it's a funny thing not a uh serious issue or serious

06:41:09
thing so my question is if debugging is removing bugs okay that's what we say right debugging is removing bugs then what is coding if debugging is removing bugs then coding is adding bugs so what I'm saying is every time time you write one line of code that is a potential bug let me show you this line can be a potential bug this line can be a potential bug it will not it may not come up when you run the application but may based on the different uh input from the user it might behave differently and

06:41:42
that's why we do testing right so every time you write your application it's important for you to test your application that's very important so one way to reduce the testing and uh one way to reduce the potential works is don't write the code write minimum lines of code what if the spr the framework which you working with says hey you know you don't have to write all these lines I will give it to you behind the scene the good thing is all this particular uh Frameworks they are well tested and they

06:42:13
are getting used in the industry from a long time so at least you can trust them that they are not creating issues okay I'm not saying it is 100% safe that there have been some ISS some uh cases where you have used certain libraries and they found some issues with the library and because of which the Hat attack happened uh you might have heard about log 4J in fact in the spring framework also there was some I forgot the library but if anyone knows let me know in the chat uh it was having some

06:42:41
issues and again it's not like the issue stays it gets rectified but uh it happens I'm not saying it is 100% safe but try to write less amount of code okay cool uh so this is what we have done in in the database but then in the yesterday session we have seen something called a spring framework Now using spring framework spring provides you something called a dependency injection which means you don't have to create the object you simply have to use the object uh example if I want the object of

06:43:12
product service in the previous example uh on day two basically we were able to create the object with the help of new keyword but the moment you use a new keyword that means you are handling the object object you creating object you are doing everything with it so in in uh spring what we doing is we are not creating object by yourself we are simply saying there's a spring container so spring Frameworks actually create a container for you imagine a container as a box which has lot of different objects

06:43:39
and you simply Say Hey I want that object example let's say if you want a phone of course you will not manufacture the phone by yourself you will simply go to a shop and buy the phone in the same way spring conna says I have all the objects which you need just ask for it it okay and that's what we doing we saying hey Spring get me a bean get me the object basically the objects in Spring containers are called Bean of course it has more logic to it but let's not talk about the logic at this point

06:44:08
imagine being as an object okay and then we are getting the object of this particular class which is uh product service and then you got the object and you can use the object for your purpose right and we able to call the methods now the beauty is there's nothing Chang change in the service right you can see we have all the same methods of we have commented it but it should it should work and uh product service needs the object of product DB right of course if you have a class which is calling

06:44:37
another class methods you need object for it and you don't have to say new keyword what you can simply do is all the classes or all the interfaces which is repository basically uh what you can do is you can write a annotation now let's not discuss the difference between repository service component again it will take another two hours just to understand it but you know just a gist basically spring will say okay the moment you mention repository the moment you mention uh service the moment you

06:45:10
mentioned component where we have used it uh entity and you can also use component what

spring will do is every class which has this annotation on top of it it will create object for you and you you don't have to get object by yourself spring will give it to you and you simply say give me the object and one way to do that is by using autowired now how exactly it is searches let's not talk about those things okay but then it will it will give it to you now what we are going to do today is we will be

06:45:39
building we will be converting this project into a web project now today's session might be B shorter because we don't have much to do uh we just have to convert this Pro into web so let's do that okay so let's get back to our drawing board so basically what we're trying to do is first of all why do we build a web project is because when it comes to standard on Project what we have done it only works on my machine if I want to use the same thing on someone else's machine basically I have to build

06:46:08
I have to take the entire project give it to them and they will run it example in our office we have four working machines and if only one machine has the software they will not be able to access it and also where will be database database is is in this machine then we have to connect and all the stuff so what we can do is instead of building a standard on Project we can build uh a web project where anyone can use it not just us we can we can use it the entire world can use it you can also use it uh maybe we can create a account

06:46:43
repository you can have your own account and you can manage your own products uh we can do that and uh okay that's one so if you talk about client you know we are poor people we have a client we have a laptop here so basically from this laptop I want to access this database see ultimately it's all about database right as a user you're not concerned about uh I mean you you are concerned about the data you want data right so how a user will be connected to a database so basically a user uh I mean

06:47:12
of course if a user wants to send a request database only works with one language which is SQL if you talk about the langu uh database like dbms so when you talk about D dbms we have uh our dbms specifically we use SQL language example we are using poiz here right now when we use poiz so that is rdbms you need to use SQL now from a user of course as a imagine a normal user a user will not be searching something online uh not be searching something by using SQL query I mean ask your parents or ask

06:47:43
your friends who are non from non it and say hey if you want to check your profile on Instagram just say select uh star from Instagram database where the username is this of course as a normal user we don't do that right U so what you get is a UI so basically a client will get a UI here so this is where the UI will be and this UI will send the request to the server of course a server can be on the same database can be on the same machine or database can be on some other machine doesn't matter but

06:48:12
let's say at this point we have on the same machine now in this machine we have to first create something here so a user will send a request right a user will send a request and what a user will get is response now specifically if we talk about a protocol which we use to send and accept the request we use something called HTTP right so we have to use HTTP protocol to send and receive

the request now this particular box here this is responsible to accept the request from the client so whatever client's request

06:48:48
it will say okay I got this request let me see if I can fulfill that or not okay and then based on that it will send the response as well yeah so based on that based on that it will send a response as well now this type of layer in between we call them as controllers now what is a job of a controller just to accept the request from the client and send the response let me repeat the only job of a controller here is to accept the request from a client and send the response now whatever extra you want to

06:49:22
do example let's say if you want want to process something client says Hey controller I want to add two numbers now question arise where you will add those two numbers will you add that in a controller or somewhere else so never remember never do any calculations in the controller controller job is to just to accept the request and send the response now after controller you will be having one more layer here which is your service layer now this layer is responsible to perform the operations

06:49:52
example let's say client says hey uh server I want to add two numbers okay server says give me the request uh I mean of course I will work only when you send the request the controller this particular controller here will accept the request and then it will say oh you are basically trying to add two numbers let me take help from the service now it is service responsibility to add those two numbers and then of course service can send the response to the controller and controller will send the response to

06:50:22
the client if that makes sense but what if client says Hey I want to know some data which which is there in the database example let's say uh you want to fetch a product which is in the database now question arise who will do it is it your service who will do it or someone else so basically we have one more layer here which is your data layer in fact uh yeah we can say data layer it has multiple names to it but just to keep it simple we have a database layer or we can say database layer uh data layer doesn't

06:50:54
matter and then this request goes to service will request to a datab database layer and database will send the response now this is how you basically have multiple layers in the web application a client sends a request request go to the controller controller sends a request to the service service send a request to the data data will have the SQL queries which will send request to the database fresh the data send it to service send it to service service will send it to controller controller will send it to the client

06:51:21
okay uh now the beauty is uh a client or or a controller can send the data to the client right but how the data should look like of course the data should be fancy right it should not be a simple uh data you want to see data on a proper layout and all the stuff so what you can do is you can send a HTML page so basically what a clients receive HTML but nowadays when you talk about clients it's not just a web page it's not just a web browser right in the earlier days we used to only use web

06:51:51

browsers but now we have multiple devices example you can have a phone uh you can have uh what else we can have any anyway let's say we have phone as well now a client here wants to access data from a phone now if you open any application a application we already have a layout right because see in a browser when you see a page you don't have a layout here you simply have a blank page whatever it has to show should be coming from the server but on the mobile phone when you talk about any application doesn't

06:52:26
matter is it Android iOS or any type of mobile phone you have a layout specified what changes is data example let's say if you go to Instagram and when you scroll basically you have those boxes fixed the data changes right so we already have a layout here now from the server side do you need a HTML page or do you just want a data of course right you just want a data now question arise what should be the format of this data normally a data can come into different formats in the earlier days we used to be we used to

06:53:01
use XML but now the famous one is you can also write in the chat the famous one is Json which is Javascript object notation so whatever data you get from the controller will be coming in the Json format now the same Json format will be uh will be showed on the Android of course the layout is fixed data is coming it will show the data and even if you're using a a browser what you can do is instead of sending an HTML file what you can do is you can send a react page a single page application and then the

06:53:34
react will already have a layout every time you do something it will send a request to the server and get the data and that's why if you see on the front end part react is very famous right and mobile you can also use Android or iOS applications I hope this makes sense everyone okay uh so now what we want to do is we basically want to create this controller now how do we do it and since we are creating a web project we also need a server so we need a server where you can run this now different languages have different

06:54:08
servers example if you want to work with PHP uh I think PHP uses a Zam server right uh we also have vam server Zam server there are entire shop of servers for PHP uh then if when it comes to okay what other the language JavaScript so if you if you are using a JavaScript in the backend uh basically you need a node there nodejs and that's why if you want to build a back end application you use node there and uh if you are using Java basically if you are using uh if you want to use a web server

06:54:44
one of the most famous one is your uh Tomcat okay so basically if you want to use the entire thing you need to install Tom Cat in your machine okay so that's what we we are trying to build here this is this this is the entire thing we're going to do now in this particular session we will not be talking about the front end okay so this section here this section here is not something we are concerned about we just want to focus on the backend part does it make sense we are focusing only on

06:55:17
this but now question arise even if you build this how will you use it if you don't have a front end how will you use it so what we can do is see see the thing is every time you have a server who says you can send a request and you can get the response even if a server provides 10 features so let's say a server says I will provide you 10 features or a server has a 10 feature but server only wants to

provide five features to the public how do we do it how do we provide the service to the

06:55:45
outside world let me repeat my question if a server has so many features or so many services how it will expose those services to the outside world and that's why we have this concept called API so basically from the server side you have to expose some apis does it make sense everyone so we basically have to uh build this API and uh expose it now question if you don't have a front end how will you expose this so that's why what we can do is instead of using this front end here we can use a API

06:45:39
building we will be converting this project into a web project now today's session might be B shorter because we don't have much to do uh we just have to convert this Pro into web so let's do that okay so let's get back to our drawing board so basically what we're trying to do is first of all why do we build a web project is because when it comes to standard on Project what we have done it only works on my machine if I want to use the same thing on someone else's machine basically I have to build

06:46:08
I have to take the entire project give it to them and they will run it example in our office we have four working machines and if only one machine has the software they will not be able to access it and also where will be database database is is in this machine then we have to connect and all the stuff so what we can do is instead of building a standard on Project we can build uh a web project where anyone can use it not just us we can we can use it the entire world can use it you can also use it uh maybe we can create a account

06:46:43
repository you can have your own account and you can manage your own products uh we can do that and uh okay that's one so if you talk about client you know we are poor people we have a client we have a laptop here so basically from this laptop I want to access this database see ultimately it's all about database right as a user you're not concerned about uh I mean you you are concerned about the data you want data right so how a user will be connected to a database so basically a user uh I mean

06:47:12
of course if a user wants to send a request database only works with one language which is SQL if you talk about the langu uh database like dbms so when you talk about D dbms we have uh our dbms specifically we use SQL language example we are using poiz here right now when we use poiz so that is rdbms you need to use SQL now from a user of course as a imagine a normal user a user will not be searching something online uh not be searching something by using SQL query I mean ask your parents or ask

06:47:43
your friends who are non from non it and say hey if you want to check your profile on Instagram just say select uh star from Instagram database where the username is this of course as a normal user we don't do that right U so what you get is a UI so basically a client will get a UI here so this is where the UI will be and this UI will send the request to the server of course a server can be on the same database can be on the same machine or database can be on some other machine doesn't matter but

06:48:12

let's say at this point we have on the same machine now in this machine we have to first create something here so a user will send a request right a user will send a request and what a user will get is response now specifically if we talk about a protocol which we use to send and accept the request we use something called HTTP right so we have to use HTTP protocol to send and receive the request now this particular box here this is responsible to accept the request from the client so whatever client's request

06:48:48
it will say okay I got this request let me see if I can fulfill that or not okay and then based on that it will send the response as well yeah so based on that based on that it will send a response as well now this type of layer in between we call them as controllers now what is a job of a controller just to accept the request from the client and send the response let me repeat the only job of a controller here is to accept the request from a client and send the response now whatever extra you want to

06:49:22
do example let's say if you want want to process something client says Hey controller I want to add two numbers now question arise where you will add those two numbers will you add that in a controller or somewhere else so never remember never do any calculations in the controller controller job is to just to accept the request and send the response now after controller you will be having one more layer here which is your service layer now this layer is responsible to perform the operations

06:49:52
example let's say client says hey uh server I want to add two numbers okay server says give me the request uh I mean of course I will work only when you send the request the controller this particular controller here will accept the request and then it will say oh you are basically trying to add two numbers let me take help from the service now it is service responsibility to add those two numbers and then of course service can send the response to the controller and controller will send the response to

06:50:22
the client if that makes sense but what if client says Hey I want to know some data which which is there in the database example let's say uh you want to fetch a product which is in the database now question arise who will do it is it your service who will do it or someone else so basically we have one more layer here which is your data layer in fact uh yeah we can say data layer it has multiple names to it but just to keep it simple we have a database layer or we can say database layer uh data layer doesn't

06:50:54
matter and then this request goes to service will request to a datab database layer and database will send the response now this is how you basically have multiple layers in the web application a client sends a request request go to the controller controller sends a request to the service service send a request to the data data will have the SQL queries which will send request to the database fresh the data send it to service send it to service service will send it to controller controller will send it to the client

06:51:21
okay uh now the beauty is uh a client or or a controller can send the data to the client right but how the data should look like of course the data should be fancy right it should not be a simple uh data you want to see data on a proper layout and all the stuff so what you can do is you can send a

HTML page so basically what a clients receive HTML but nowadays when you talk about clients it's not just a web page it's not just a web browser right in the earlier days we used to only use web

06:51:51
browsers but now we have multiple devices example you can have a phone uh you can have uh what else we can have any anyway let's say we have phone as well now a client here wants to access data from a phone now if you open any application a application we already have a layout right because see in a browser when you see a page you don't have a layout here you simply have a blank page whatever it has to show should be coming from the server but on the mobile phone when you talk about any application doesn't

06:52:26
matter is it Android iOS or any type of mobile phone you have a layout specified what changes is data example let's say if you go to Instagram and when you scroll basically you have those boxes fixed the data changes right so we already have a layout here now from the server side do you need a HTML page or do you just want a data of course right you just want a data now question arise what should be the format of this data normally a data can come into different formats in the earlier days we used to be we used to

06:53:01
use XML but now the famous one is you can also write in the chat the famous one is Json which is Javascript object notation so whatever data you get from the controller will be coming in the Json format now the same Json format will be uh will be showed on the Android of course the layout is fixed data is coming it will show the data and even if you're using a a browser what you can do is instead of sending an HTML file what you can do is you can send a react page a single page application and then the

06:53:34
react will already have a layout every time you do something it will send a request to the server and get the data and that's why if you see on the front end part react is very famous right and mobile you can also use Android or iOS applications I hope this makes sense everyone okay uh so now what we want to do is we basically want to create this controller now how do we do it and since we are creating a web project we also need a server so we need a server where you can run this now different languages have different

06:54:08
servers example if you want to work with PHP uh I think PHP uses a Zam server right uh we also have vam server Zam server there are entire shop of servers for PHP uh then if when it comes to okay what other the language JavaScript so if you if you are using a JavaScript in the backend uh basically you need a node there nodejs and that's why if you want to build a back end application you use node there and uh if you are using Java basically if you are using uh if you want to use a web server

06:54:44
one of the most famous one is your uh Tomcat okay so basically if you want to use the entire thing you need to install Tom Cat in your machine okay so that's what we we are trying to build here this is this this is the entire thing we're going to do now in this particular session we will not be talking about the front end okay so this section here this section here is not something we are concerned about we just want to focus on the backend part does it make sense we are focusing only on

06:55:17

this but now question arise even if you build this how will you use it if you don't have a front end how will you use it so what we can do is see see the thing is every time you have a server who says you can send a request and you can get the response even if a server provides 10 features so let's say a server says I will provide you 10 features or a server has a 10 feature but server only wants to provide five features to the public how do we do it how do we provide the service to the

06:55:45
outside world let me repeat my question if a server has so many features or so many services how it will expose those services to the outside world and that's why we have this concept called API so basically from the server side you have to expose some apis does it make sense everyone so we basically have to uh build this API and uh expose it now question if you don't have a front end how will you expose this so that's why what we can do is instead of using this front end here we can use a API

06:56:19
tool a tool where you can send request to the API and one of the most most famous API tool is Postman there are multiple options but then Postman is very famous so let's use it cool let's now it's time to get started with the project okay let's re-watch the session then if this complex yeah you have to watch it if you if this is the first time you're seeing this it's very complex uh so do rewatch it okay let's get back to the code and what we're going to do here is we will

06:56:51
build a web application right and of course we can reuse the code here and what I want to do is I just want to close the earlier project so that we'll have less windows open we already have so many things going on and uh stuff I hope you're also practicing what we are doing in the session because just watching a session will not make any sense uh you will forgot this concept in a day or two so do practice this yeah behind the scene behind the scene even if you're using a spring framework

06:57:20
behind the scene this are your suets controllers are your cets uh if you want to go detail of spring mvy I will I will I will make sure that there's a link in description you will get entire idea how this controllers are created uh because see what we are going to see today is spring boot okay and spring boot actually makes your work very very easy but the thing is if you really want to understand what is happening behind the scene that's why you have to understand how spring framework works so spring

06:57:47
boot is actually a layer on top of spring framework to make your work easy so what I will do is I will create a spring web project now how do we do that just go to the browser and the same website which we have went yesterday which is spring initialiser select Maven select Java all the same settings which we have done uh project name here product where spring web because we are building a web project now and jar file v file okay now the difference between jar and war is in V basically you uh you deploy the

06:58:21
project on a tomcat server so basically the place where you are deploying it should have a tomcat server jar what it does is it embeds the Tomcat server inside the project that means if you have a project ready you can just run it because Tom CAD is there already uh okay now what about dependencies we need Post gisp in this particular uh thing as well we also need data jpa because that's how we going to connect with database apart from this we need one more thing which is

web okay so we need a spring web

06:58:53
what else we need let me just check am I missing something here I think spring web is what we need uh we can also add lombok here because I want to reduce this code let me see if I have lombok I've never tried lombok in this yeah we have Lok here cool now once you have all these things click on generate your project will be ready to use let's unzip it done and let's open this project so go back to intell idea file uh open uh downloads spring product spring web click on open uh new window because we

06:59:37
want to use the earlier window as well and voila you got your spring web project ready okay now as I mentioned the Tomcat is already present in this particular project if I go back to if I go to external external libraries and if you search here Tom Cat will be already present where are you Tom cat oh don't tell me they are giving you glassfish now oh yeah it's here so you can see we got Tom Kat I don't know why I was not able to see that properly oh he it here uh okay says can I also

07:00:09
win this MacBook I watch all your live stream for sure hes uh but I think you already got one from Apple yeah H H says it so you already have a Mac Mac Studio so you can see Tom is already already here so what I will do is I will just minimize this and let's create a project now or we already have a project let's expand it now if you check out the pom.xml now this is where you have all the dependencies right uh so we got dat jpa we got spring web we got postgress we got the lombok as well to reduce the

07:00:41
code and apart from this if you go back to SRC and if you see the files we already have one file here which is the main file and if you expand this no fancy code and you can see we got okay we have one more file here which is application properties which we'll see later but we got the project right now if you are coming from the old world of java let's say if you are talking about suret or spring framework normal spring MVC we need to do lot of configuration even before you run the project let me run this project what I

07:01:14
will do is even before running the project let me go back to my browser and here let me search for Local Host colon 880 enter and you can see it says this site cannot be reached because the server is not running so if you want to access your Tomcat with a default port number now this is a default port number for Tomcat and you can see this is not running what I will do is I will run this application without installing any tomcat and stuff because Tomcat is there embedded if IUN this code okay oh the

07:01:47
same problem I don't know why I add the data jpa because the the the thing is springboard is very smart now since we have added data jpa it is searching for the database connections you know I just to keep it happy what I will do is I will just get those connections which we have done yesterday all these connections copy and paste in the application. properties which is empty in this case at least let's keep our uh data jpa happy right we have done that and now let's run this once again

07:02:20
let's see what happens run and you can see it is running the Tom Cat started on port 8080 so just

by downloading the project you are able to run it no external dependencies that's fun right uh let's go back here refresh can you see that yeah we got the error because that's our life but at least you got the error from the Tomcat it says the homepage which you're searching for is not there because we are accepting the request but then the request which which is coming from the client we not able to process

07:02:59
it because there's no mapping there right okay so what we'll do is uh let's do the mapping now question arise where to do the do the mapping so I will just just go back here and create a controller remember in this we have talked about a controller so we need to get this controller here so I will say new and let's create a class and let's name this as a product oh I'm not sharing my screen let me do that again what I've done so right click on the package say new and say

07:03:37
Java class and click on product controller so we got a product controller here now how do you make this a controller this is a normal class right how do you make this as a controller we have to give give it some extra power how do you give that to extra power how do you make this a controller just by having a controller in your name will not make you a controller so how do you make this a controller it's very simple you simply add something called a rest controller now question arise controller makes

07:04:09
sense why it is called rest see there's a concept of rest API now since we are uh yeah now since we are using uh HTTP request if you want to do anything on the server side if you want to uh get the data if you want to save data I mean send data if you want to update data if you want to uh delete data what you can do is you can use something called rest which stands for repres representational State transfer whatever communication happening happening between client and a server it's all about transferring the

07:04:44
states right sending the values and that's why we say a rest controller to know more about rest controller I will put a a video in description this will make much more sense then so you can see we got a r controller and now this is a controller it is ready to accept the request but the thing is if you talk about a web application you have multiple pages right uh so you have multiple pages and every page will have a different address if you see if You observe the address bar your uh address bar changes for different request in the

07:05:13
same way for this particular application as well you have to specify what you want to do uh do do you want to uh get the products do you want to add a product do you want to update a product do you want all the products do you want a product based on a particular uh name for that you have to provide multiple mappings example if I go back to my browser and if I say okay from this page not the courses I want all the products I can simply say Local Host colon 880 slash products the moment I enter it's

07:05:45
not working because there's no mapping happening so what I will do is for this particular request for this particular request what I can do is I can create a method so if someone is calling products I have to call this method now this method returns a list of products right because you're asking for all the products so list of products uh we have to okay we don't have that class yet so I will say get products or I can say get all products because we are fetching all the products that's it the thing is

we don't

07:06:16
have this class product here can you see that we don't have this class so what to do okay so basically uh we have a class here which is a product and we don't have a class so what we'll do is we'll go back to earlier project and copy this class product okay not here here let's space this class so we got product now there's no issue we have to import the package for list okay and then here we can simply return okay what do we return oh I don't know how to F this value it's

07:06:49
because controller will not do all those things right controller needs helps from another layer with what is that layer yeah that's right we want these two classes service and dat database copy this and paste because we need these two layers as well the service layer and the DB layer so what you can do is you can create an object of a product service so you can say product service I will say service and should I say new keyword no right I can simply inject it so the way you can inject that is with

06:56:19
tool a tool where you can send request to the API and one of the most most famous API tool is Postman there are multiple options but then Postman is very famous so let's use it cool let's now it's time to get started with the project okay let's re-watch the session then if this complex yeah you have to watch it if you if this is the first time you're seeing this it's very complex uh so do rewatch it okay let's get back to the code and what we're going to do here is we will

06:56:51
build a web application right and of course we can reuse the code here and what I want to do is I just want to close the earlier project so that we'll have less windows open we already have so many things going on and uh stuff I hope you're also practicing what we are doing in the session because just watching a session will not make any sense uh you will forgot this concept in a day or two so do practice this yeah behind the scene behind the scene even if you're using a spring framework

06:57:20
behind the scene this are your suets controllers are your cets uh if you want to go detail of spring mvy I will I will I will make sure that there's a link in description you will get entire idea how this controllers are created uh because see what we are going to see today is spring boot okay and spring boot actually makes your work very very easy but the thing is if you really want to understand what is happening behind the scene that's why you have to understand how spring framework works so spring

06:57:47
boot is actually a layer on top of spring framework to make your work easy so what I will do is I will create a spring web project now how do we do that just go to the browser and the same website which we have went yesterday which is spring initialiser select Maven select Java all the same settings which we have done uh project name here product where spring web because we are building a web project now and jar file v file okay now the difference between jar and war is in V basically you uh you deploy the

06:58:21
project on a tomcat server so basically the place where you are deploying it should have a tomcat server jar what it does is it embeds the Tomcat server inside the project that means if you have a

project ready you can just run it because Tom CAD is there already uh okay now what about dependencies we need Post gisp in this particular uh thing as well we also need data jpa because that's how we going to connect with database apart from this we need one more thing which is web okay so we need a spring web

06:58:53
what else we need let me just check am I missing something here I think spring web is what we need uh we can also add lombok here because I want to reduce this code let me see if I have lombok I've never tried lombok in this yeah we have Lok here cool now once you have all these things click on generate your project will be ready to use let's unzip it done and let's open this project so go back to intell idea file uh open uh downloads spring product spring web click on open uh new window because we

06:59:37
want to use the earlier window as well and voila you got your spring web project ready okay now as I mentioned the Tomcat is already present in this particular project if I go back to if I go to external external libraries and if you search here Tom Cat will be already present where are you Tom cat oh don't tell me they are giving you glassfish now oh yeah it's here so you can see we got Tom Kat I don't know why I was not able to see that properly oh he it here uh okay says can I also

07:00:09
win this MacBook I watch all your live stream for sure hes uh but I think you already got one from Apple yeah H H says it so you already have a Mac Mac Studio so you can see Tom is already already here so what I will do is I will just minimize this and let's create a project now or we already have a project let's expand it now if you check out the pom.xml now this is where you have all the dependencies right uh so we got dat jpa we got spring web we got postgress we got the lombok as well to reduce the

07:00:41
code and apart from this if you go back to SRC and if you see the files we already have one file here which is the main file and if you expand this no fancy code and you can see we got okay we have one more file here which is application properties which we'll see later but we got the project right now if you are coming from the old world of java let's say if you are talking about suret or spring framework normal spring MVC we need to do lot of configuration even before you run the project let me run this project what I....give important and teach me like this only in very very simple

Okay Samiksha, ab full simple mode ON 😄
Main sirf **important points** pick karungi, bilkul easy language mein, step by step.

1️⃣ ApplicationContext & getBean – “Spring se object maangna”

In Spring Boot main class:

```java
ApplicationContext context =
        SpringApplication.run(ProductSpringApplication.class, args);
```

Iska simple meaning:

- Spring ek **dabba (container)** banata hai
- Is dabbe ke andar saare objects (beans) ready rakhta hai (ProductService, ProductRepository, etc.)

Ab agar tumko ProductService ka object chahiye:

```java
ProductService service = context.getBean(ProductService.class);
```

Iska matlab:

> "Spring, mujhe `ProductService` ka object de de."

Ye hi **dependency injection** hai:

- Pehle: `new ProductService()` tum banati thi
- Ab: Spring banata hai, tum sirf **maangti ho**

2️⃣ JPA Repository – pura ProductDB class gayab 🎩✨

Pehle tum aise likh rahi thi:

- `ProductDB` class
- usme `save()`, `getAll()`, SQL queries, `PreparedStatement`, `ResultSet` etc.

Spring Data JPA ke baad:

```java
public interface ProductDB extends JpaRepository<Product, Integer> {
}
```

Bas **ek line**, aur:

- `findAll()` – saare products
- `save(product)` – naya product save
- `deleteById(id)` – delete
- `count()` – kitne rows

Sab methods **already ready** mil jaate hain.
Tum SQL likhti hi nahi. Spring hidden mein query likh ke chala deta hai.

3️⃣ @Repository + @Autowired – "Spring, ye wala object inject kar de"

```java
@Repository
public interface ProductDB extends JpaRepository<Product, Integer> { }
```

`@Repository` ka simple matlab:

> “Spring, isko **database layer** samajh, iska object bana ke container mein rakh.”

Fir Service ke andar:

```java
@Autowired
private ProductDB db;
```

`@Autowired` ka matlab:

> “Spring, mujhe **`ProductDB`** ka object chahiye, please yaha inject kar de.”

Tum **`new ProductDB()`** nahi kar rahi ho.
Spring khud object de raha hai.

4️⃣ application.properties – DB ka address, username, password

Yaha tum Spring ko batati ho:
“Database kahan hai aur kaise connect hona hai?”

```properties
spring.datasource.url=jdbc:postgresql://localhost:5432/telescope
spring.datasource.username=postgres
spring.datasource.password=0000
```

Bas itna:

- **`url`** – DB kis jagah hai + kaunsa DB (PostgreSQL)
- **`username`** – kaun login karega
- **`password`** – uska password

Baaki connection, pooling, sab Spring handle karega.

5️⃣ Product class ko Entity banana – table se mapping

Pehle normal POJO tha:

```java
public class Product {
    private int id;
    private String name;
    private String type;
    private String place;
    private int warranty;
}
```

Ab Spring Data JPA ke liye:

```java
@Entity
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    // name, type, place, warranty + getters/setters
}
```

- **`@Entity`** → ye class DB ki **table** se map hogi (**`product`** table)
- **`@Id`** → ye column **primary key** hai
- **`@GeneratedValue(...)`** → ID **auto-increment** hogi (1,2,3,...)

Ab jab tum:

```java
db.save(product);
```

call karogi:

- Spring **`insert into product(...)`** query khud bana ke DB mein daal dega

6️⃣ Layers ka simple picture (Web project mein)

Socho tumhari app 4 floors wali building hai:

1. **Client (Browser / Mobile / Postman)**
2. **Controller** – request lene/dene wala
3. **Service** – logic karne wala
4. **Repository (DB layer)** – DB se baat karne wala

Flow:

> Client → Controller → Service → Repository → Database
> Database → Repository → Service → Controller → Client

Controller example:

```java
@RestController
public class ProductController {
    @Autowired
    private ProductService service;

    @GetMapping("/products")
    public List<Product> getAll() {
        return service.getAllProducts();
    }
}
```

- `@RestController` → ye HTTP request handle karega
- `/products` URL pe jab koi hit karega
- Controller → `ProductService` ko bolega → wo `ProductDB` (repository) se data lega

Response JSON mein jaata hai, jo browser / React / Android sab samajh sakte hain.

7️⃣ Spring Web + Postman + REST idea

- Hum **Spring Web** dependency add karte hain → HTTP APIs bana sakte hain
- Ye APIs ko hum **REST API** bolte hain
- Tool like **Postman** se tum direct APIs test kar sakti ho:
  - GET `/products`
  - POST `/products`
  - etc.

Front-end na ho tab bhi tum apna backend test kar sakti ho.

8️⃣ Spring Initializr – project ready karwana

Website: **start.spring.io**

Waha bas choose:

- Project: Maven
- Language: Java
- Dependencies:
  - Spring Web
  - Spring Data JPA
  - PostgreSQL Driver
  - Lombok (optional, boilerplate kam karne ke liye)

Download → unzip → IntelliJ/Eclipse mein open → run.
Tomcat already embedded hota hai, alag se install nahi karna.

9️⃣ Ek chhota summary in your style 💛

- Spring Boot = **helper** jo Tomcat, config, sab ready rakhta hai
- ApplicationContext = **dabba** jisme saare objects (beans) ready hote hain
- `getBean()` / `@Autowired` = “Spring, object de de”
- Spring Data JPA = “SQL main likhungi hi nahi, tu kar de”
- `JpaRepository<Product, Integer>` = free me `findAll`, `save`, `delete` etc.
- `@Entity`, `@Id`, `@GeneratedValue` = class ↔ table mapping
- `application.properties` = DB ka address, username, password
- `@RestController` = HTTP request handle karega, JSON return karega
- Layers: Controller → Service → Repository → DB

If you want, next step mein main:

- **poora Product example** Spring Boot + JPA + REST ke saath
- full flow code dungi: `Product`, `ProductRepository`, `ProductService`, `ProductController`, `application.properties`
- har file ka **2–3 line simple explanation**

Bas bolo:

> “Haan, ab full code flow bhi simple way mein dikhao.”

07:01:14
will do is even before running the project let me go back to my browser and here let me search for Local Host colon 880 enter and you can see it says this site cannot be reached because the server is not running so if you want to access your Tomcat with a default port number now this is a default port number for Tomcat and you can see this is not running what I will do is I will run this application without installing any tomcat and stuff because Tomcat is there embedded if IUN this code okay oh the

07:01:47
same problem I don't know why I add the data jpa because the the the thing is springboard is very smart now since we have added data jpa it is searching for the database connections you know I just to keep it happy what I will do is I will just get those connections which we have done yesterday all these connections copy and paste in the application. properties which is empty in this case at least let's keep our uh data jpa happy right we have done that and now let's run this once again

07:02:20
let's see what happens run and you can see it is running the Tom Cat started on port 8080 so just by downloading the project you are able to run it no external dependencies that's fun right uh let's go back here refresh can you see that yeah we got the error because that's our life but at least you

got the error from the Tomcat it says the homepage which you're searching for is not there because we are accepting the request but then the request which which is coming from the client we not able to process

07:02:59
it because there's no mapping there right okay so what we'll do is uh let's do the mapping now question arise where to do the do the mapping so I will just just go back here and create a controller remember in this we have talked about a controller so we need to get this controller here so I will say new and let's create a class and let's name this as a product oh I'm not sharing my screen let me do that again what I've done so right click on the package say new and say

07:03:37
Java class and click on product controller so we got a product controller here now how do you make this a controller this is a normal class right how do you make this as a controller we have to give give it some extra power how do you give that to extra power how do you make this a controller just by having a controller in your name will not make you a controller so how do you make this a controller it's very simple you simply add something called a rest controller now question arise controller makes

07:04:09
sense why it is called rest see there's a concept of rest API now since we are uh yeah now since we are using uh HTTP request if you want to do anything on the server side if you want to uh get the data if you want to save data I mean send data if you want to update data if you want to uh delete data what you can do is you can use something called rest which stands for repres representational State transfer whatever communication happening happening between client and a server it's all about transferring the

07:04:44
states right sending the values and that's why we say a rest controller to know more about rest controller I will put a a video in description this will make much more sense then so you can see we got a r controller and now this is a controller it is ready to accept the request but the thing is if you talk about a web application you have multiple pages right uh so you have multiple pages and every page will have a different address if you see if You observe the address bar your uh address bar changes for different request in the

07:05:13
same way for this particular application as well you have to specify what you want to do uh do do you want to uh get the products do you want to add a product do you want to update a product do you want all the products do you want a product based on a particular uh name for that you have to provide multiple mappings example if I go back to my browser and if I say okay from this page not the courses I want all the products I can simply say Local Host colon 880 slash products the moment I enter it's

07:05:45
not working because there's no mapping happening so what I will do is for this particular request for this particular request what I can do is I can create a method so if someone is calling products I have to call this method now this method returns a list of products right because you're asking for all the products so list of products uh we have to okay we don't have that class yet so I will say get products or I can say get all products because we are fetching all the products that's it the thing is we don't

07:06:16
have this class product here can you see that we don't have this class so what to do okay so basically uh we have a class here which is a product and we don't have a class so what we'll do is we'll go back to earlier project and copy this class product okay not here here let's space this class so we got product now there's no issue we have to import the package for list okay and then here we can simply return okay what do we return oh I don't know how to F this value it's

07:06:49
because controller will not do all those things right controller needs helps from another layer with what is that layer yeah that's right we want these two classes service and dat database copy this and paste because we need these two layers as well the service layer and the DB layer so what you can do is you can create an object of a product service so you can say product service I will say service and should I say new keyword no right I can simply inject it so the way you can inject that is with

07:07:30
the help of Auto wired and we got it we got the service class and now with this service you will say service dot get all products job done you simply converted your normal project to a web project voila this is your web project now but will this work that's the real question let's try let's restart the server and there is no error you can see go back to the browser refresh oh it's not working it is not working what's wrong what's wrong there my friend what's wrong the thing is we created

07:08:14
this method but nowh there's a mapping right of course a controller can have multiple methods different methods should execute for different requests so we have to mention in English hey this is for products request simple right no but that's a comment right how jav will understand that uh so what we can do is we can say add it to do that we have to say get mapping and in this bracket you have to mention what is your mapping so this will be for products SL products so whenever you get request for

07:08:53
products this is what you want to execute save go back okay we have to restart the server restart no issue refresh wo we got it is this making sense okay how to copy files is very simple just go back to the earlier project and just copy the files right click copy and paste in your recent project now whatever you're building in fact I'm not following a proper standards here you know the controller should go to controllers package uh the DB goes to the repo package uh service goes to service

07:09:27
package product goes to model package and that's why we say MVC right which is model view controller so the data which is passing between all these layers are called models right product is a model here yeah springboard is actually interesting cool cool cool you know why I'm not creating those packages because the project is already so complex that's a good practice you should always do that but yeah I'm teaching you the concept so it works for me okay now what about uh I want to send

07:10:00
the data okay I want to create new entry in the database how do we do that how do we send the new recorder that's a question now let's say if you have a form on your mobile application or a web page you can enter data and you can say send but I don't have a UI here right how will I send data from here and what should be the request now the request can be I want to send a product not products because when you are fetching it you are fetching the products right but what if you want

to

07:10:32
send a product one how do we do that in fact you know before that I just want to show you one more thing what if you want to fetch one particular product not multiple uh so let's say if I want to fetch okay all let's say I want to fetch Black based okay so what I can do is in the request I can say hey I don't want all the products I want one product that's why product slash I want Black based enter this is not working how to make it work it's very simple you can add one more uh mapping copy and paste

07:11:10
and here instead of saying get all products I can say get product and it is not returning a list it should return a product simple product and the mapping will be product slash Black based right and then you cannot return all the products you have to return one product so you can say service. get okay but the problem is I think we have commented it in the service class so if I go back to Service uh you can see we have commented this and the thing is this should be coming from the database not from uh

07:11:48
list because we have not coded this in the last session let's me let me code it fast so I can simply say return DB dot uh find by okay if you have to find by name uh name right because you can use find but then it will work with uh the primary key you can see we got find by ID uh if you want to do find by we do have find by but will this work oh we have to pass the query function as well so what I can do is uh spring J data jpa in fact data jpa provides you one more thing you can go to database and if you want

07:12:24
to fetch from particular column not from the primary key uh what you can do is you can say return a product and find by name so whatever column name you have so say find by this is fixed what will change is the column name so find my name and whatever name you get here string name and the beauty is you don't even have to Define this okay you don't even have to Define this okay where do you find the the last session uh go to the channel and there's a question where do you find the last

07:12:58
question let me just do that so go to the channel and uh search for a playlist tab so in the playlist there's a 10day chall challenge you can follow the videos there okay just add this method go back to your service and say DB do find by name and whatever your name is no need to Define no need to write any uh SQL query or something now go back to the controller and say get product by name which is uh okay where's the name let me just try with uh black beast I'm hardcoding it at this point

07:13:38
let's see if this works blackbeast restart the server of course you can also install Dev tools which will help you to directly restart the server every time you make a changes okay let's refresh this oh it works you can see we got Black based but the thing is it will only work for Black based if you if you try to use some other name it will not work you know why is because we have hardcoded the values we are saying black beas black Bas at both the places so what you can do is instead of saying

07:14:12

black black beast you can put that in curly brackets and say name so whatever data you passing from there is saying this is so much easier than day three content okay see this is easy because we have done this in day three if could have skipped the day three this will this is like next level complex because we have not even talked about this layers here so we have done the difficult part in the last session okay so you can see so whatever data you're getting you will save that in this particular name in The

07:14:42
Curly brackets and then you can accept it in here so you can say name so whatever name is coming here you can accept it here okay let's try and the same variable can be used here now instead of saying black Bas because see it doesn't matter what you search it will always give you Black based we don't want it we don't we want to so whatever user says here whatever user says here that's what you want to accept so restart and refresh oh it's not working you can see it's not working

07:15:09
what went wrong I was expecting error but there's no error give me something okay so let me tell you what the what a problem is the thing is you have to use one annotation here which is path variable if you want to use this so by path variable what you're saying is in the path whatever variable you have just try to map it so you can see name name try to map it that was missing there and now refresh it works you can use it for any other name uh in fact I remember we have Asus Rog right that's one machine

07:15:39
but the thing is since we have a space between two words uh on the internet if you want to use web address web URL you have to use percentage 20 which is a code for space just say enter and you can see we got Asus Rog is it making sense and the format which you can see here is dis format okay Patrick sounds more like uh giloy from Silicon Valley okay so let me go back to my code again and now let's do it for the adding a product because we were able to fetch products what if you want to add a

07:16:09
product okay so if you want to do that what you can do is uh you can create public and since we are adding it we can use void because we don't want anything in return normally you know a good practice is to to return the status code uh HTTP has different status code example 4004 is for error related to resources 500 is the other related to the servers uh 200 is okay everything is going well so every time you perform some operation you have to return the status code as well at in this particular session we are not doing

07:16:42
it because again it would be very complex okay so public void uh add product and now question arise what what should be a part of this parenthesis or sorry the round brackets if you are passing a single value you will accept that in a string now you are passing an object from the client side how will you do it so since we are sending data from a client we have to send the entire object of a product because that's what we are adding so we have to accept that here so we have to say product P so we are accepting it the

07:17:15
entire product we are accepting it now once you accept it uh we can simply say service dot add okay we don't have ADD method okay we have to implement so many things now uh in this service this is the add method okay product and t. save okay this should work let's go back to the controller say add product pass p and your job is done but how do you map it now question arise

what type of request you are sending so basically whenever we talk about HTTP we have different protocols there or different methods there in

07:17:56
HTTP so if you want to fetch data we use get and when you want to save data or when you want to send data we use post okay so we have to use post mapping here not get mapping so let me repeat when you want to get data we use when you want to fresh data we use get when you want to send data we use post and what should be the URL I want to be consistent with a product we'll say/ product now you will be saying don't you think the this one and this one matches this product and this product they are matching but then look

07:18:37
at the mapping as well we got get mapping we got post mapping here okay of course this is also different but even if they are same we got two different types of mapping there one is get and second is post mapping now people who are coming from this s background they know what I'm talking about right do get do Post okay so we got this right but the question is will this work let's go back to our browser and let's send the data oh but how do I send data from here okay because see browser supports

07:07:30
the help of Auto wired and we got it we got the service class and now with this service you will say service dot get all products job done you simply converted your normal project to a web project voila this is your web project now but will this work that's the real question let's try let's restart the server and there is no error you can see go back to the browser refresh oh it's not working it is not working what's wrong what's wrong there my friend what's wrong the thing is we created

07:08:14
this method but nowh there's a mapping right of course a controller can have multiple methods different methods should execute for different requests so we have to mention in English hey this is for products request simple right no but that's a comment right how jav will understand that uh so what we can do is we can say add it to do that we have to say get mapping and in this bracket you have to mention what is your mapping so this will be for products SL products so whenever you get request for

07:08:53
products this is what you want to execute save go back okay we have to restart the server restart no issue refresh wo we got it is this making sense okay how to copy files is very simple just go back to the earlier project and just copy the files right click copy and paste in your recent project now whatever you're building in fact I'm not following a proper standards here you know the controller should go to controllers package uh the DB goes to the repo package uh service goes to service

07:09:27
package product goes to model package and that's why we say MVC right which is model view controller so the data which is passing between all these layers are called models right product is a model here yeah springboard is actually interesting cool cool cool you know why I'm not creating those packages because the project is already so complex that's a good practice you should always do that but yeah I'm teaching you the concept so it works for me okay now what about uh I want to send

07:10:00
the data okay I want to create new entry in the database how do we send the

new recorder that's a question now let's say if you have a form on your mobile application or a web page you can enter data and you can say send but I don't have a UI here right how will I send data from here and what should be the request now the request can be I want to send a product not products because when you are fetching it you are fetching the products right but what if you want to

07:10:32
send a product one how do we do that in fact you know before that I just want to show you one more thing what if you want to fetch one particular product not multiple uh so let's say if I want to fetch okay all let's say I want to fetch Black based okay so what I can do is in the request I can say hey I don't want all the products I want one product that's why product slash I want Black based enter this is not working how to make it work it's very simple you can add one more uh mapping copy and paste

07:11:10
and here instead of saying get all products I can say get product and it is not returning a list it should return a product simple product and the mapping will be product slash Black based right and then you cannot return all the products you have to return one product so you can say service. get okay but the problem is I think we have commented it in the service class so if I go back to Service uh you can see we have commented this and the thing is this should be coming from the database not from uh

07:11:48
list because we have not coded this in the last session let's me let me code it fast so I can simply say return DB dot uh find by okay if you have to find by name uh name right because you can use find but then it will work with uh the primary key you can see we got find by ID uh if you want to do find by we do have find by but will this work oh we have to pass the query function as well so what I can do is uh spring J data jpa in fact data jpa provides you one more thing you can go to database and if you want

07:12:24
to fetch from particular column not from the primary key uh what you can do is you can say return a product and find by name so whatever column name you have so say find by this is fixed what will change is the column name so find my name and whatever name you get here string name and the beauty is you don't even have to Define this okay you don't even have to Define this okay where do you find the the last session uh go to the channel and there's a question where do you find the last

07:12:58
question let me just do that so go to the channel and uh search for a playlist tab so in the playlist there's a 10day chall challenge you can follow the videos there okay just add this method go back to your service and say DB do find by name and whatever your name is no need to Define no need to write any uh SQL query or something now go back to the controller and say get product by name which is uh okay where's the name let me just try with uh black beast I'm hardcoding it at this point

07:13:38
let's see if this works blackbeast restart the server of course you can also install Dev tools which will help you to directly restart the server every time you make a changes okay let's refresh this oh it works you can see we got Black based but the thing is it will only work for Black based if you if you try to use some other name it will not work you know why is because we have hardcoded the

values we are saying black beas black Bas at both the places so what you can do is instead of saying

07:14:12
black black beast you can put that in curly brackets and say name so whatever data you passing from there is saying this is so much easier than day three content okay see this is easy because we have done this in day three if could have skipped the day three this will this is like next level complex because we have not even talked about this layers here so we have done the difficult part in the last session okay so you can see so whatever data you're getting you will save that in this particular name in The

07:14:42
Curly brackets and then you can accept it in here so you can say name so whatever name is coming here you can accept it here okay let's try and the same variable can be used here now instead of saying black Bas because see it doesn't matter what you search it will always give you Black based we don't want it we don't we want to so whatever user says here whatever user says here that's what you want to accept so restart and refresh oh it's not working you can see it's not working

07:15:09
what went wrong I was expecting error but there's no error give me something okay so let me tell you what the what a problem is the thing is you have to use one annotation here which is path variable if you want to use this so by path variable what you're saying is in the path whatever variable you have just try to map it so you can see name name try to map it that was missing there and now refresh it works you can use it for any other name uh in fact I remember we have Asus Rog right that's one machine

07:15:39
but the thing is since we have a space between two words uh on the internet if you want to use web address web URL you have to use percentage 20 which is a code for space just say enter and you can see we got Asus Rog is it making sense and the format which you can see here is dis format okay Patrick sounds more like uh giloy from Silicon Valley okay so let me go back to my code again and now let's do it for the adding a product because we were able to fetch products what if you want to add a

07:16:09
product okay so if you want to do that what you can do is uh you can create public and since we are adding it we can use void because we don't want anything in return normally you know a good practice is to to return the status code uh HTTP has different status code example 4004 is for error related to resources 500 is the other related to the servers uh 200 is okay everything is going well so every time you perform some operation you have to return the status code as well at in this particular session we are not doing

07:16:42
it because again it would be very complex okay so public void uh add product and now question arise what what should be a part of this parenthesis or sorry the round brackets if you are passing a single value you will accept that in a string now you are passing an object from the client side how will you do it so since we are sending data from a client we have to send the entire object of a product because that's what we are adding so we have to accept that here so we have to say product P so we are accepting it the

07:17:15

entire product we are accepting it now once you accept it uh we can simply say service dot add okay we don't have ADD method okay we have to implement so many things now uh in this service this is the add method okay product and t. save okay this should work let's go back to the controller say add product pass p and your job is done but how do you map it now question arise what type of request you are sending so basically whenever we talk about HTTP we have different protocols there or different methods there in

07:17:56
HTTP so if you want to fetch data we use get and when you want to save data or when you want to send data we use post okay so we have to use post mapping here not get mapping so let me repeat when you want to get data we use when you want to fresh data we use get when you want to send data we use post and what should be the URL I want to be consistent with a product we'll say/ product now you will be saying don't you think the this one and this one matches this product and this product they are matching but then look

07:18:37
at the mapping as well we got get mapping we got post mapping here okay of course this is also different but even if they are same we got two different types of mapping there one is get and second is post mapping now people who are coming from this s background they know what I'm talking about right do get do Post okay so we got this right but the question is will this work let's go back to our browser and let's send the data oh but how do I send data from here okay because see browser supports

07:19:05
fetching of course you can use some tools internally to do that but if you want to send data let's use a real tool which test which will use to test the apis and that's why we have Postman now Postman is specialized in uh request sending and response what I will do is I will start with the products so let's try this first this address in the postman this is how Postman looks like if you want to download Postman you can go to post man.com and you can download depending upon your OS so go back to

07:19:35
postman and send set a get request get request for this particular URL which is Local Host colon 880 products and says send and you can see you will get all your data here and this is a status code you can you got there but if you you can customize it from the server side uh let's try it for product/ blackbeast send and you can see we only got black beast but now I want to send request I want to send a post request so you will go to post the URL is this which is Local Host colon 8/ product but now this time we

07:20:08
have to send data as well how do you send data so what you can do is you can just copy this now if you want to send data you will do that in a body so in the HTTP body you do that and replace this with the and let's change values so we already have till 15 let's add 16 one of the item on my table is uh a pen pen is of type pen it's on white table expiry but never expires let's say 9999 Okay so Yeah so basically you have to use yeah we can also use headers actually yeah but then let's let's use

07:20:59
body uh you can set the raw and you can set the data like this you can set the format what you're sending you're sending Json and say okay the pro the problem is uh yeah it should work we can't we don't have to set the value right for ID let's try let's try says send okay it says product not found that's weird here we forgot to rest the server is it yeah ID is actually autogenerated so we don't

have to pass the ID but let's say even if you're passing it no harm run

07:21:30
something happened you can see there's no error you got 200 let's send the get request again for the products send you can see we got all the products if you go down oh oh it's not working it's not working you can see the data is not assigned it's I mean we were able to send the request data is also saved in the database but not the data which we have sent okay it was not able to accept data that's the problem so we have to use one more anation here which is request body so you have to mention hey

07:22:02
I'm sending something from the body accept it okay uh so we have to explain everything to Spring framework okay let's try now let's send this data again because this time it will use another ID run oh sorry we have to send a post request in fact where is my history here I'm using Postman for a long time yeah here so let me use a post request with the same body send nothing went wrong here let me use the get request send go back yeah can you see that we got it so we got pen a white

07:22:43
table and everything is this making sense oh that's great okay now uh I want to show you one more thing about lombok right how do we how do we integrate Lok here okay and now it's time for lombok let's go to lombok Lok how do we integrate that we have already added the longbo part in the uh in the XML and where do we want to use it I want to use it for the product if you look at the product class there are so many things here right what you can do is for this particular variables I don't

07:23:19
want to create gets sets you can skip the entire part from here even two string you can say I don't want to write it it it makes your code so lengthy right uh so what you can do is you can just go on top of your class and say data and that's it it will autogenerate the geta status for you so basically lombok is used to reduce the number of lines of code you write in a class welcome back alens my name is ready and finally the Java project is here that's right I know you waiting for this from a

07:23:51
long time uh so basically after watching Code Java after watching spring framework there was one question right what next how do we build a project and that's where I have a small project for you so basically we'll do this project together using spring Boot and we are going to use a database as well for data storage right and for this project we are going to use mongod Deb now why spring boot why mongodb will discuss that bit later and also we need a front end right and for front end we are going

07:24:18
to use react now basically basically in this project we are not going to build reite application let's say we already have the UI ready now we have to work on the back end part so the backend part will be done with the help of spring Boot and the database will be mongodb even before looking at the project there is one particular question in your mind and that is is it a good time to watch this video maybe you don't know spring boot maybe you don't know mongod DB how will you understand this project so I am

07:24:44
making this video considering two situation the first one is what if you know spring Boot and Mong DB the basics you are at the right place is because you understand how will you use your

knowledge of spring boat and build an application and also if you know mongodb Basics you will know how to use the mongodb cloud which is atlas now what if you don't know this Technologies and that's the beauty now I'm making sure that when I'm explaining you the concepts at least you will understand

07:25:12
the flow of a project you will understand why I'm doing certain things and why exactly that line of code but yeah if you are if you want to do it by yourself after watching the videos that's where you will get stuck but of course there's different way of learning stuff sometime you have to learn from the basics and sometime you have to come from top to bottom okay this is what you can do in the project and now let me learn the basics so doesn't matter at what stage you are you can enjoy these

07:25:37
sessions now let me talk about the project which we're going to build so the project is actually very simple this is a project basically we are building a platform for someone to get a job and what if someone want to hire someone so basically this portal or this website is good for both the employer and for the job Seeker let's say you are a jobseeker and you are looking for a job I know this is a very basic UI but again as I mentioned right the idea of this video is not to understand the UI part the

07:26:05
main thing is the back end so I I wanted some UI so we have it here and you will find this entire project the UI the back end in the description so you will get a GitHub link and you you can access the project and you can run on your machine okay so you don't have to type everything but by yourself don't worry okay so let me click on this get job now so basically I'm looking for a job and these are all the job description which I have so these people are looking for candidates who knows so let's say if I

07:26:31
talk about the first one they're looking for developer and basically they're looking for software engineer okay I've just typed the description the way I want it doesn't matter you know the grammar is not may not be perfect the technology might not match we're just here to learn right so let's go with that so we got some description looking for software engineer who can work with spring Boot and mongodb and react okay you are watching the right project then uh the experience required is one

07:26:54
year it should be year but then that's fine uh these are the skill set required in terms of Technology uh then we have the same profile let's say if you go scroll down we also have Junior data scientist uh spelling mistake here but you can ignore uh so data scientist who has end to endend project experience that's great 12 years I'm not even sure if data science is that old but anyway it's my data I can do whatever I want with that okay so skill set required Python numai and those

07:27:20
stuff okay so this looks cool right and also we have an requirement for editor who knows Final Cut Pro and who can use my Mac Studio uh the actual editor is not using it so I'm looking for someone else okay doesn't matter so let's say these are the profiles we have here and now let's say if I have lot of job requirements let's say 1,000 or 10,000 how do I search for my specific requirement of course you can use a search box here so in the search box of course you can search for a particular

07:27:48
profile let's say I want to search for developer so I can type developer here okay and it should give me the results so it is giving me all the results of developers you can see developer developer developer developer here but what if I'm not sure what profile I need but I'm sure that I know something example let's say I know I know threads so I can simply type I know threads and you can see it will find a profile where the skill set is threads okay so basically we are also able to search and

07:28:15
that's exciting right because you're implementing a search feature and trust me if you're do it by yourself using a Java code or any programming language it is very time consuming and we have to write a lot of lines of code so we'll understand how do we do this using the mongodb address so here let's say if I want to search for maybe I'm looking for a job only in Mumbai so I can type Mumbai you can say it will give you the jobs which in the description where Mumbai is mentioned so basically for

07:28:40
sure when you search for something we not just searching in a profile so let's say this is a profile here this is a description this is the number of years of have experience wir and these are the Technologies you should know so basically we are not just searching in the profile we are also searching for the description and the technology okay maybe it's like a more of a Google search which we are trying to build it will be quite awesome right and also what if you want to hire someone maybe

07:19:05
fetching of course you can use some tools internally to do that but if you want to send data let's use a real tool which test which will use to test the apis and that's why we have Postman now Postman is specialized in uh request sending and response what I will do is I will start with the products so let's try this first this address in the postman this is how Postman looks like if you want to download Postman you can go to post man.com and you can download depending upon your OS so go back to

07:19:35
postman and send set a get request get request for this particular URL which is Local Host colon 880 products and says send and you can see you will get all your data here and this is a status code you can you got there but if you you can customize it from the server side uh let's try it for product/ blackbeast send and you can see we only got black beast but now I want to send request I want to send a post request so you will go to post the URL is this which is Local Host colon 8/ product but now this time we

07:20:08
have to send data as well how do you send data so what you can do is you can just copy this now if you want to send data you will do that in a body so in the HTTP body you do that and replace this with the and let's change values so we already have till 15 let's add 16 one of the item on my table is uh a pen pen is of type pen it's on white table expiry but never expires let's say 9999 Okay so Yeah so basically you have to use yeah we can also use headers actually yeah but then let's let's use

07:20:59
body uh you can set the raw and you can set the data like this you can set the format what you're sending you're sending Json and say okay the pro the problem is uh yeah it should work we can't we don't have to set the value right for ID let's try let's try says send okay it says product not found

that's weird here we forgot to rest the server is it yeah ID is actually autogenerated so we don't have to pass the ID but let's say even if you're passing it no harm run

07:21:30
something happened you can see there's no error you got 200 let's send the get request again for the products send you can see we got all the products if you go down oh oh it's not working it's not working you can see the data is not assigned it's I mean we were able to send the request data is also saved in the database but not the data which we have sent okay it was not able to accept data that's the problem so we have to use one more anation here which is request body so you have to mention hey

07:22:02
I'm sending something from the body accept it okay uh so we have to explain everything to Spring framework okay let's try now let's send this data again because this time it will use another ID run oh sorry we have to send a post request in fact where is my history here I'm using Postman for a long time yeah here so let me use a post request with the same body send nothing went wrong here let me use the get request send go back yeah can you see that we got it so we got pen a white

07:22:43
table and everything is this making sense oh that's great okay now uh I want to show you one more thing about lombok right how do we how do we integrate Lok here okay and now it's time for lombok let's go to lombok Lok how do we integrate that we have already added the longbo part in the uh in the XML and where do we want to use it I want to use it for the product if you look at the product class there are so many things here right what you can do is for this particular variables I don't

07:23:19
want to create gets sets you can skip the entire part from here even two string you can say I don't want to write it it it makes your code so lengthy right uh so what you can do is you can just go on top of your class and say data and that's it it will autogenerate the geta status for you so basically lombok is used to reduce the number of lines of code you write in a class welcome back alens my name is ready and finally the Java project is here that's right I know you waiting for this from a

07:23:51
long time uh so basically after watching Code Java after watching spring framework there was one question right what next how do we build a project and that's where I have a small project for you so basically we'll do this project together using spring Boot and we are going to use a database as well for data storage right and for this project we are going to use mongod Deb now why spring boot why mongodb will discuss that bit later and also we need a front end right and for front end we are going

07:24:18
to use react now basically basically in this project we are not going to build reite application let's say we already have the UI ready now we have to work on the back end part so the backend part will be done with the help of spring Boot and the database will be mongodb even before looking at the project there is one particular question in your mind and that is is it a good time to watch this video maybe you don't know spring boot maybe you don't know mongod DB how will you understand this project so I am

07:24:44
making this video considering two situation the first one is what if you know spring Boot and Mong

DB the basics you are at the right place is because you understand how will you use your knowledge of spring boat and build an application and also if you know mongodb Basics you will know how to use the mongodb cloud which is atlas now what if you don't know this Technologies and that's the beauty now I'm making sure that when I'm explaining you the concepts at least you will understand

07:25:12
the flow of a project you will understand why I'm doing certain things and why exactly that line of code but yeah if you are if you want to do it by yourself after watching the videos that's where you will get stuck but of course there's different way of learning stuff sometime you have to learn from the basics and sometime you have to come from top to bottom okay this is what you can do in the project and now let me learn the basics so doesn't matter at what stage you are you can enjoy these

07:25:37
sessions now let me talk about the project which we're going to build so the project is actually very simple this is a project basically we are building a platform for someone to get a job and what if someone want to hire someone so basically this portal or this website is good for both the employer and for the job Seeker let's say you are a jobseeker and you are looking for a job I know this is a very basic UI but again as I mentioned right the idea of this video is not to understand the UI part the

07:26:05
main thing is the back end so I I wanted some UI so we have it here and you will find this entire project the UI the back end in the description so you will get a GitHub link and you you can access the project and you can run on your machine okay so you don't have to type everything but by yourself don't worry okay so let me click on this get job now so basically I'm looking for a job and these are all the job description which I have so these people are looking for candidates who knows so let's say if I

07:26:31
talk about the first one they're looking for developer and basically they're looking for software engineer okay I've just typed the description the way I want it doesn't matter you know the grammar is not may not be perfect the technology might not match we're just here to learn right so let's go with that so we got some description looking for software engineer who can work with spring Boot and mongodb and react okay you are watching the right project then uh the experience required is one

07:26:54
year it should be year but then that's fine uh these are the skill set required in terms of Technology uh then we have the same profile let's say if you go scroll down we also have Junior data scientist uh spelling mistake here but you can ignore uh so data scientist who has end to endend project experience that's great 12 years I'm not even sure if data science is that old but anyway it's my data I can do whatever I want with that okay so skill set required Python numai and those

07:27:20
stuff okay so this looks cool right and also we have an requirement for editor who knows Final Cut Pro and who can use my Mac Studio uh the actual editor is not using it so I'm looking for someone else okay doesn't matter so let's say these are the profiles we have here and now let's say if I have lot of job requirements let's say 1,000 or 10,000 how do I search for my specific requirement of course you can use a search box here so in the search box of course you can search for a particular

07:27:48
profile let's say I want to search for developer so I can type developer here okay and it should give me the results so it is giving me all the results of developers you can see developer developer developer developer here but what if I'm not sure what profile I need but I'm sure that I know something example let's say I know I know threads so I can simply type I know threads and you can see it will find a profile where the skill set is threads okay so basically we are also able to search and

07:28:15
that's exciting right because you're implementing a search feature and trust me if you're do it by yourself using a Java code or any programming language it is very time consuming and we have to write a lot of lines of code so we'll understand how do we do this using the mongodb address so here let's say if I want to search for maybe I'm looking for a job only in Mumbai so I can type Mumbai you can say it will give you the jobs which in the description where Mumbai is mentioned so basically for

07:28:40
sure when you search for something we not just searching in a profile so let's say this is a profile here this is a description this is the number of years of have experience wir and these are the Technologies you should know so basically we are not just searching in the profile we are also searching for the description and the technology okay maybe it's like a more of a Google search which we are trying to build it will be quite awesome right and also what if you want to hire someone maybe

07:29:06
if we want to post a new job so let's say in my office I need a requirement of a blockchain developer so I can click on hire talent and let's say job profile I will say blockchain developer and number of experience required is two let's say say a description person should be familiar with ethereum developer who knows how to make dabs that's it and then the the requirement is let's say the person should know JavaScript anyway we don't have a option of solidity here but again I wanted to keep it light so

07:29:37
we got JavaScript there you can change the UI the way you want it's all about the submission which you do for the server so when I click on submit and if I search for let's say blockchain developer now is it there it should be okay let me refresh so you can see if I go back to this page if I go back to get job now we have a requirement for blockchain developer cool right and if I search for dab okay you can see we got the developer here okay so this is what we want to build now for this of course the

07:30:10
react app is ready so I will show you the code a bit but we not be writing the react code by ourself but we will be requiring a spring boot so we are going to build this with spring Boot and we are going to use use mongodb in the back end so let's get started so we'll minimize this screen I I don't want this and in fact I wanted to show you the vs code so this is basically the the uh the react project which we have so you can see we have multiple files here this is for the form this is for the feeds what

07:30:35
you see all the developers all the job posting and you can see the request is getting sent to the

server again if you don't want to use react in the front end you can you can use normal uh UI as well or even if you don't want to touch the UI part I'm not a big fan of UI by the way we can use the apis calling maybe you can use Postman to test these apis or we can use Swagger okay so we'll use all those things in this project I know it's quite exciting so let's get started

07:31:02
okay so what we'll do is first of all let's create a project from scratch now what are the Technologies we going to use here so we are going to use inj for uh Java we are going to use the database which is mongodb now why I'm choosing mongodb here why not other database see the thing is there's a huge debate debate between SQL and nosql datab databases which are good I have made a video as well before this so you can find the link in description which talks about what is nosql so basically just to

07:31:30
give you an idea there are different types of databases available we can categorize them as SQL and non-sql in SQL we create tables right so we have a database in database we have tables so in in nosql basically we don't maintain the structure now why it is important is because when SQL got famous the at that point it was around 1970s at that point the memory was very costly so we have to make sure that we use minimum storage and we can store maximum data and that's why the SQL type of table makes sense

07:32:00
now if you talk about the current world we have so much of data which is unstructured you can't actually create the columns based on that so what if your table is created and now you want to add a new column trust me it's a big headache in SQL that's one so basically we don't want to use SQL here the second one is you want to scale your application you want you want to have replicas of your of your database it is very difficult to do it in SQL of course you can have replicas but then it's very

07:32:26
difficult to sync it so not no SQL database excel in that and we can Implement that with the help of mongodb so mongodb is a nosql database where you don't have tables what you have is documents so if you want to relate it's very easy database database is there on both the side in SQL we use tables right which is which has data here we have collections so we can have tables related to collections in SQL you store your data in the row format right so if you want to represent a particular

07:32:52
entity let's say I want to store my information my name my age my phone number my email ID that becomes one row now that in nosql becomes a document now the way we can have multiple rows in SQL we can have multiple documents which is a part of a collection that makes sense right so basically whatever data you're storing in fact the data which you have here this is a profile this is a description this is the experience and this is a technology you know so this becomes one document and if you have

07:33:22
multiple data that multiple documents okay and we don't have to install any software for mongodb because we are going to use the cloud service and trust me this skill sets which we are going to do in today's session is very important okay so we I already have a intj on my machine which is int Community version we are going to use that but then we want to create a spring project right and since we have a community version I'm not paying for it so we have to make sure that we have that

second we want to create a service

07:33:48
we want to create a mongodb service right so basically what we need to do is first we need to get the account on mongodb Atlas okay it's very simple you can just click on it and the important thing is it will not ask you for credit card information hopefully let's see if that uh is asking for you uh so I will sign up with my Google account and this should be a quick process so click on submit you can choose your email ID and you can choose you can click on submit I normally prefer to have Google login

07:34:13
everywhere so that I don't don't have to remember the passwords so it says welcome you uh use your account to deploy depl Loy a cloud database okay that's what we're going to do and it's still building it okay so basically it is asking you some questions you can just enter okay we are here to learn uh mongodb whatever application we are building we are building content management and then preferred language so basically we are going to do that with Java now that's a beauty about

07:34:37
mongodb okay which is you can use multiple languages here now since we are going to use Java so I can click on Java here and click on finish now if you're getting confused how exactly the document looks like so if you have ever worked with Java JavaScript the JavaScript object you know how you create that's how the document is getting stored so basically that's a Jon format for us but in the storage it is getting stored as Bon which is binary sun right which is Javascript or we can say binary Json to make it easy okay

07:35:03
they still loading the page maybe the internet is low on my side and that's what it is we got our story the only thing is it will ask you what type of service you want okay so if you can go for serverless you can go for dedicated but you have to pay for it now since we are into learning stage we can select the free one so you can you can click on share and you can click on create it will create a cluster for you now of course this is not a point where we talk about cluster imagine you're getting

07:35:28
these services and the beauty is you can choose any of these cloud service so mongod D behind the scene will allow you to choose AWS or Google cloud or Azure so we are going to use AWS here okay so region is Mumbai by default you're selecting the good things uh don't need to change anything here the only thing I want to change here is the cluster name I will name this cluster as the go and click on create cluster the next thing is it will ask you for the username and password to protect your database right

07:35:57
so if you want to connect you have to enter the username password so you can see we have the root uh by default I've enter entered this before as well so it is taking the default values so you can enter whatever data you want to have I'm just typing root and root and I will click on create user so my user is created quite simple right uh the next thing I want to do is I will be using this from the the local environment so I'm connecting my application which is running on my machine with the mongodb....give important and teach me like this only in very very simple

Samiksha,
don't worry.
Main tumhare liye poora Spring Boot + REST + JPA + Postman + MongoDB explanation **ultra-simple language** mein doongi —
bilkul wohi style jaisa video mein teacher bol raha tha… **but 10× easier, clear points** mein.

🌟 PART 1 — SPRING BOOT PROJECT RUN KA BASIC FLOW

(VERY VERY SIMPLE VERSION)

✅ 1. Localhost:8080 ka error — why?

When you type:

```makefile
localhost:8080
```

Aur **server run hi nahi ho raha**, toh browser bolega:

❌ *"This site can't be reached"*
Because **Tomcat server is not running**.

❗ But Spring Boot ke project mein Tomcat already included hota hai

Tumko Tomcat install nahi karna.
Bas **Run** → **Spring Boot Application**.

🌟 PART 2 — WHY SPRING BOOT WAS NOT STARTING?

Teacher ne bola:

> "Since we added Data JPA, Spring Boot is searching for database connection."

Simple words mein:

🔹 JPA dependency lagayi → Spring Boot sochta hai:
"Mujhe database ka connection dijiye, warna main chalega nahi."

Isliye tumne application.properties mein DB ka URL + username + password diya.

Uske baad:

✔️ Tomcat started on port 8080
✔️ Server successfully started

🌟 PART 3 — CONTROLLER BANAYE WITHOUT UI

(VERY SIMPLE WAY)

🔥 Step 1: Create a Controller Class

```java
@RestController
public class ProductController {
}
```

RestController means:

- Ye class **HTTP requests** handle karegi
- Ye data return karegi (JSON), HTML nahi

🌟 PART 4 — URL MAPPINGS (GET, POST)

1️⃣ GET request for ALL products

```java
@GetMapping("/products")
public List<Product> getAllProducts() {
    return service.getAllProducts();
}
```

Meaning:

> "Agar koi `/products` hit karega... toh mujhe **saare products return** karne hain."

2️⃣ GET request for ONE product — using Path Variable

URL:

```bash
/product/BlackBeast
```

Code:

```java
@GetMapping("/product/{name}")
public Product getProduct(@PathVariable String name) {
    return service.getProductByName(name);
}
```

💡 *`{name}` means URL se value uthao.*

🌟 PART 5 — SERVICE LAYER (BUSINESS LOGIC)

Service is like a helper:

```java
@Service
public class ProductService {

    @Autowired
    ProductDB db;

    public List<Product> getAllProducts() {
        return db.findAll();
    }

    public Product getProductByName(String name) {
        return db.findByName(name);
    }

    public void addProduct(Product p) {
        db.save(p);
    }
}
```

🌟 PART 6 — REPOSITORY LAYER

(THIS IS THE MAGIC OF SPRING DATA JPA)

```java
@Repository
public interface ProductDB extends JpaRepository<Product, Integer> {
    Product findByName(String name);
}
```

Important:

- **`extends JpaRepository`** → CRUD free mil jata hai
- **`findByName`** → Spring khud query generate karega
- No SQL, no statements, no boilerplate

🌟 PART 7 — ENTITY (Product Table Mapping)

```java
@Entity
@Data
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String name;
    private String type;
    private String place;
    private int warranty;
}
```

Very simple:

- `@Entity` → ye DB table hai
- `@Id` → primary key
- `@GeneratedValue` → auto increment
- `@Data` (Lombok) → getter/setter/tostring banata hai

🌟 PART 8 — POSTMAN USE FOR ADDING DATA

POST request

URL:

```bash
http://localhost:8080/product
```

Select:

```css
Body → raw → JSON
```

Send JSON:

```json
{
  "name": "Pen",
  "type": "Stationery",
  "place": "White Table",
  "warranty": 9999
}
```

Controller Code:

```java
@PostMapping("/product")
public void addProduct(@RequestBody Product p) {
    service.addProduct(p);
}
```

👉 **@RequestBody** means:
“whatever JSON is coming from Postman → convert to Product object”

🌟 PART 9 — WHY REQUEST BODY IMPORTANT?

Without:

```java
@RequestBody
```

Spring **unable to read JSON**, so data → null.

With:

✔ Converts JSON → Java Object
✔ Saves into DB

🌟 PART 10 — WHY PATHVARIABLE IMPORTANT?

Without:

```java
@PathVariable
```

Spring does NOT know:

- URL ka **`{name}`** kis variable mein daalna hai

With:

```java
@GetMapping("/product/{name}")
public Product getProduct(@PathVariable String name)
```

✔ Automatically maps URL value → method variable

🌟 PART 11 — MONGODB SIMPLE EXPLANATION

SQL → tables, rows, columns
MongoDB → collections, documents, fields

Example Mongo Document

```json
{
  "profile": "Developer",
  "description": "Looking for engineer",
  "experience": 1,
  "techStack": ["Java", "Spring", "MongoDB"]
}
```

No fixed columns
No ALTER TABLE
No migrations
Super flexible!

🌟 PART 12 — ATLAS MONGO SETUP (VERY EASY)

1. Go to: **mongodb.com/atlas**
2. Create account
3. Select **FREE cluster**
4. Region → Asia/Mumbai
5. Create username & password (root/root)
6. Add IP Address: **0.0.0.0/0**
7. Get connection string
8. Paste in Spring Boot:

```bash
spring.data.mongodb.uri=mongodb+srv://root:root@cluster/database
```

Done 🎉

❤️ Samiksha, THIS WAS THE SIMPLEST BREAKDOWN

If you want next:

✔️ Full real BACKEND project

✔️ Full flow diagram explained like kids

✔️ Full CRUD code in simplest form

✔️ JSON tests using Postman

✔️ MongoDB queries very simple

✔️ Spring Boot project structure explanation

✔️ Complete revision of all concepts

Just type:

“Teach me full Spring Boot project step by step very simple.”

I’ll make a complete guide for you.