1) Tell us about yourself

Can you tell us about yourself or can you talk about your work experience while walking us through your résumé?

No matter what the question is and how it is asked, this ice breaking question is a great opportunity to market our skills and all the great work we have been doing over the years.

It is like fishing , where we will leave several technical baits hoping that the interviewer will be caught in one or most of them and ask us questions from those topics as the interview progresses.

This is the very first question we will be responding to, and it conveys a lot about our communication, confidence and experience.So we should prepare six to 12 points and practice those points 30 to 40 times, at least

2) About Me Preparation Template

I'll walk you through some of the important points from my about me, which I have been using over the years, I'll always start by saying I have started my career as a Java developer in 2002.

Since then, I have worked across domains ranging from ERP to CRM banking to finance, project management to healthcare and travel to gaming.

This point tells them my overall experience and also my experience across various domains, which can be very important for certain roles.

Next, I move on to the technology side where I state while working across the Java technology full stack, I have designed and developed applications using various spring boot modules and also frontend javascript framework like Angular and React.

So depending on whether it is a full stack position or if it is a backend position, you can change this point slightly.

And based on your experience, whether you are a junior or senior developer etc, you can add more technologie here if you want to.

Next, I have worked closely with the product owners and business analysts to get the requirements right and with the architects to implement the non-functional requirements like security, scalability, reliability,etc.This point makes you a real good professional at the same time it also showcases that you are not just a simple developer, but you can interpret the requirement and also, if required, you can work with the architects to implement the non functional requirements.
Again, if you are a junior developer, you can take out the nonfunctional side of things if you are not into it yet.Next, we'll move on to the devops side of things where I state

I have also worked with devops engineers to set up the CI/CD pipeline for our project using tools like Maven,Docker, Jenkins' and Kubernetes.

So these are all the baits we are leaving them so that they can ask us more questions on these later on.So we should be thorough in those topics.In some cases, we may not have the devops team. We ourselves will set up this whole thing. So depending on your project, you can change this point accordingly.But the key here, state your devops experience and your CI/CD experience and the tools you have used.

I have also worked with various aws components, such as EC2 , S3, EKS and so on, to deploy our apps to the cloud. You can modify this to the cloud you have worked with pretty much every job description has a cloud requirement today.

So with which cloud you are familiar with, you can use that here instead of aws so you can replace it with Azure GCP, etc..

Some projects that do not need cloud, they might be using Kubernetes for their deployment or they are still migrating to the cloud.

In such cases, you can skip it if you do not have that experience.But if you do have that experience, please state all that in detail.

Moving on to the agile side of things, I have participated in several Agile conferences and played a key role in implementing Scrum and other agile practices like tdd pair programming, etc. So depending on your experience, you can state your agile experience here. You can say how you have followed the scrum ceremonies, the planning meetings and all that.You can put it into one point, which is very important.As soon as you get onto a team, they expect you to immediately become agile and be a part of their agile process that they are following last and very important.

I am a continuous learner. I have recently learned how to create serverless projects using AWS lambdas.You can state whatever you have learned recently as this last point because teams need continuous learner as the industry evolves, as new things evolve.And also, I also like sharing what I learn and I do it through my YouTube channel and blog.This has helped me tremendously over the years.Not many people share from what I see in the industry, so when you love to share.People love it because they need mentors.Once you get onto the team, they allow you to share what you are learning on a continuous basis and also learning from others as well.

So whatever you are doing, please add those points at the end. And if you do not have a blogger YouTube channel yet, please do create one right away so you can use this as a template and create one of your own because you are unique.

3) Can you talk about your recent project

Always remember that it is another great opportunity to highlight and sell the skills that you have used on your recent project for the position you are being interviewed for .

Start with the high level use case of the problem you are trying to solve.

For example, if you are working on a flight reservation and check in application, we can tell them that our application allows the end users to book their flight tickets and check into the flights online or through their mobile applications.

You then draw the high level architecture diagram, which will show them the various components of your application.

Feel free to use the whiteboard if it is a Face-To-Face interview. If there is a whiteboard available, grab the whiteboard with their permission, of course, and draw the high level architecture diagram and show the flow in between these components.

If a whiteboard is not available, use a white sheet of paper.If it is a virtual meeting, feel free to share your screen and paint the picture.This will leave a lasting impression on them.

As you showcase your presentation skills, you can then talk a little more about how you have implemented these components.

If it is completely a Java based full stack development, talk about it.If the backend is in Java and the Frontend is developed using other JavaScript frameworks, talk about them as well.

You can then jump into the layered architecture of your Java back end which typically will have the data access layer services layer where the business logic leaves the presentation layer and the integration layer where you can use restful web services or messaging.

You can also talk about each of these layers and the technologies that you have used in these layers.This is very important and this is where you can highlight those technologies that are really required for the job on hand.

For the data access layer typically will use spring data jpa with hibernate for the restful web
services layer.

We would have used spring boot web or frameworks like Jersey and the list goes on and on.So we have a lot of technologies which fall into these layers.

This is where your responsibilities on your resume under your current project will come in.So go through all those responsibilities and highlight those portions which are required for the current job opening, which you are being interviewed for.

Don't stop with just the development side of things.Talk on how you how written unit tests using unit test frameworks like Junit, Mockito, etc., how we have reported the test coverage of those unit tests, the servers to which you have deployed your application to.

If you did that, if you have used messaging, talk about the messaging provider or the broker you have used and if you have developed the front end using Angular react etc do talk about that as well.

Once you are done with the development side of things, move on to the devops side of things.This is where you highlight your skills, working with one of the build tools like

Maven, Gradle,etc. Talk about how you Dockerized your project, how you have created the docker file or the docker compose file for the application components within your architecture.

And if the job requirement needs kubernetes, highlight that as a container orchestration tool.

How you have deployed your containers to the kubernetes cluster if it needs aws or if your last project has used one of these clouds, highlight that as well.

Keep the conversation interactive, Don't let it be a one side affair.

You can always stop and ask them if they want to know more details of any of these as you mention them,especially if they want to hear more about how you have implemented these layers at a very low level.

If they say yes, you can walk them through the classes and interfaces which you typically create in a Java application, starting with the model classes, then the data access layer classes, as the service layer classes, the restful and Web controllers, the utility classes in your application, the validators views and many more.

This will give them a lot of confidence that you are a very hands on guy and you are ready to start coding whenever it is required.

Also, do talk about how you have implemented the non-functional requirements or the ILities, as we call them, the security, scalability, reliability, adaptability when it comes to your project.
If not, they will ask you these questions later on anyways.They will ask you how did you secure your recent project or what are the security best practices? How do you ensure that your application is scalable?And all those questions will come in throughout the interview.

CORE JAVA

1) What are the various important components that are involved in compiling and running a Java program?

When we start working on java applications, the very first software that we download and install is a JDK, which stands for Java Development Kit.

The JDK is composed of JRE and it also comes with several comments or programs such as the Java compiler,the Java command that we use to run a Java program and more.

We download the JDK and install it for a particular operating system.There are different versions for different operating systems for the JDK and the JRE, we can install JRE separately as well, but usually we install the JDK and the JRE.

When we compile the Java program, the Java compiler will convert the Java class into bytecode, not the machine code, which the underlying operating system understands, but the byte code, which is platform independent, the bytecode is only understood by the JVM.

The Java Virtual machine, which we launch using the Java Command.When we run our Java class file that gets generated during compilation with Java Command, it will launch a JVM that can interpret the bytecode to the underlying operating system that makes Java platform independent.

A JVM is an instance of the JRE.It is a representation of the Java runtime environment. So the platform independency comes because we can compile a Java program on a Windows operating system and then we can take the classes and run them on Linux using the JVM on Linux, because the bytecode is understood by the interpreter or the JVM on Linux, and it can convert it into the machine code,which Linux understands.

JIT, which stands for Just in Time compiler, is another important component, which is a part of the JVM that will interpret the code to the underlying operating system in an optimized fashion.
It knows much more details about the operating system and it converts the bytecode a little bit at a time by default, if we don't use the JIT interpreter or compiler, the JVM will translate the entire bytecode into machine code and it will execute against the operating system.

Whereas if we use the JIT, it converts a little bit of bytecode at a time and then it can do certain optimizations by inlining functions, etc., which makes our application perform better.

By default, the JIT is enabled when we use a Java command. If you want to disable it for debugging reasons or whatever, you can use the hyphen D option javadot compiler is equal to none.This will disable the JIT.

2) What are constructors

What are constructors? How do they differ from other methods of the class? Can we invoke one constructor from another constructor and can we invoke the superclass constructor from the child class constructor?

Let's take a look. A constructor is a method that is used to initialize the properties of an object when it is created. It has the same name as the class name, unlike the other methods in this case, Class Accord , the constructor is also Accord.

But other methods start and stop, have different names of their own, but the constructor name is the same name as the class name.

The constructor is only invoked when an instance or the object of that class is created, whereas the other methods can be invoked as many times as we want.

To invoke other constructor's in the same class, we used this method and we can pass in the parameters just like any other method.

And if the class accord extends Class Honda to invoke the super classes constructor the hondas constructor from within the class constructor, we use the super method and

pass in any parameters.

3) abstract class vs interface

What is the difference between an abstract class and an interface? If a class has at least one abstract method, then that class needs to be marked as abstract class, if not the class won't compile any class that extends abstract class and should then provide the implementation for
these abstract methods in an abstract class.

If not, that class should also be marked as an abstract class.Abstract class can have any number of methods with implementation, but as soon as it has one abstract method, it becomes an abstract class.

Whereas in interfaces all the methods are abstract. The class that implements the interface should provide the implementation for those abstract methods. A class can implement any number of interfaces, whereas it can only extend one abstract class.

4) Why is multiple inheritance no supported

If multiple inheritance is not supported in Java, multiple inheritance is not supported because if we inherit from two classes, which have the same method, for example, here we have the rich confused kid class, which extends father and mother, which have the same method of money.

Now, when we invoke the money method using the rich, confused child's instance or object, the compiler will not know which money method it should bind to or which money means that it should invoke.

That is one reason why multiple inheritance is not supported will cause a compile time issue.If we try to do this to make things even more complex, we have the classic diamond problem that we see in languages like C++.

If we have a grandparent class which also has a money method, then the compiler

will find it even more difficult to decide which money method should be invoked when the rich confused state object is used to invoke the money method.

5) Can a class implement two interfaces with the same method signature

Yes, here I have an interface called Car, which has the go method, which is void and also a avoid stop method.And I also have a driverless interface with sit down and relax method and a void go method, the same go method we have in the car interface as well.

Now the Honda class implements both the car and the driverless. It has no issues because it has to provide an implementation for the go method.

So it overrides the go method and it provides the implementation.So we don't have the typical Diemen problem, which we see in case of extending classes.

6) What are the Object class methods

What are the different methods that every class inherits from the object class, the different methods that every class inherits from the object class are equals hashcode finalize clone and two string.

We also have to wait, notify, etc. Those cannot be overridden, but we can use those methods in multithreading.

But these methods equals hashcode finalize.Clone and two strings can be

overridden.

7) What is the Default hashCode implementation

If we do not override the hashcode method and provide implementation for it in our class, what is the value that will be returned when we create objects of that class and invoke the hash code method?

The default hashcode method in the object class returns a value, a number that is nothing but the memory location of that particular object.

So if we do not provide our own Hashcode method, the value that will come back is the memory location of that object.

8) What is the default toString implementation

If we do not override the two string method in our classes and if we try to print an object of that class, then we will see a value which will have the class name, followed by the @ symbol and the hexadecimal representation of the memory location of that particular object.

So that is the default implementation in the object class.The two string method is implemented in such a way that it will always return the class name followed by

@ symbol and then the hexadecimal representation of the object's address.

And if we override it, then we will see whatever implementation we provide, whatever we return back from the two string method that will be used instead.

9) equals method vs == operator

Another important interview question is the differences between the equal two operator, the
double equals operator and the equals method when we compare objects.

Let's take a look at the double equal to operator by creating two objects of user.Here we are creating two instances, U1 and U2 of a class called user.

And setting the IDs to one and John, both of them have the same I.D. and name, but they have their own memory locations on the heap.

Now, if we use the double equal to operator to compare these two objects, it will return a boolean false because the double equals compares the object references or the memory locations and not the contents within the object.

So you can call it a shallow comparison and not a deep comparison Now, if you use the equals method U1 dot equals U2 the equals method is available for every Java class from the object class.

Every class implicitly extends the object class and the equals method is derived or inherited from the object class.

The default implementation of the equals method uses the double equals operator.So it is a shallow comparison by default.

So even this one, this statement U1 dot equals U2 will return a false.We have to override the equals method in our classes, the user class, and provide the implementation to compare the id and name.

There is an exception for string's primitives and enums, for the wrapper types strings and enums the equals method is already overridden to do a deep comparison.

For example, if we create two strings with the same contents ABC and ABC. S1 double equals S2 will return false because we are using a new operator to create a string.

In both cases, they both have two different memory locations.This will return a false, but if we use the S1 dot equals S2, it will do a deep comparison because the equals method is overridden on a string class and it will return a True.

Similarly for the wrapper types int I1, equal to one, two, three, integer two is equal to one, two,three.

If we do an I1 double equals I2 it will return a false.But if we do I1 dot equals I2 It will return a true.

To summarize, the double equals operator compares or looks for the references or compares
the references of two objects by default, the equals method also does the same.

We should override the equals method and do a deep comparison and return boolean value on our own for the string primitive types and enums.The equals method will do a deep comparison of the contents.

10) What are the usages or differences between final finally and finalize?

Final is a key word that when we mark variables, for example, primitive types, they become constants.We cannot do this once we define int i is equal to one, two, three as final.

The value of I cannot be changed if we use it against an object declaration.The object reference cannot change later on.We cannot point the object to a different memory location.When used against a class, the class cannot be inherited and when used against a method, we cannot override that method in a child class.

Finally is a block when we do exception handling, we use it along with try catch, we can also use finally without a catch directly, with try. Finally gets executed even when there is an exception and also and there is no exception, it gets executed always.

So it's a good place to put all the clean up code where we can clean up all the DB connections.The socket connections are input output files, streams.

Starting Java 7, we have a try with resource block, which, when used, automatically closes all the resources that are declared inside the resource block.

So starting Java 7, if we use the try with resource block, then we don't need the finally.Finally, we have the finalized method, which the JVM calls when the garbage collector is about to be called, but we should not rely on this to clean up all our resources in the program or application because we never know when the JVM is going to call the garbage collector.

11) What are generics and what is type erasure?

Prior to Java one point five version, this is how we defined a collection here, have a list into which I want to store all the employee I.D's., which are integer values.when I do this, it is fine, I'm adding an integer even when I do this, where I'm trying to add a string accidentally, the name of the employee that is allowed as well, because there is no way we are mentioning the type.

That is where Generics came in starting Java version One point five.Using generics,

we can specify or abstract out that type of data that can go into the collection.Now, if we try to add an integer, no compile time issues, if I try to add a string, the compiler will catch it right at compile time and throw an error.
Type Erasure is the process of the compiler erasing this type for runtime, once it does, it checks if the compiler will erase this type because of compatibility reasons.

Because we want the versions, the older version code probably the code written on Java one four or Java one three to run on the new JVM's

So that is the reason for backward compatibility reasons runtime Generics has been compromised or runtime generics are not present in Java.

It is all compile time Generics. So type erasure implements compile time Generics by erasing the type for runtime so that the older versions of Java code, which is compiled on JDK one four or one three, can still run on the newer JVMS.

COLLECTIONS

1) What are some of the important interfaces and classes in the collections API?

We have a list set queue, a blocking queue and map. list allows duplicates and the implementation classes of a list interface are array list and linked list. set does not allow duplicates.We have hashset linked hashset.

We also have a sorted set that is implemented by tree set, which sorts all the objects or data that is stored into that data structure.

Then we have first in, first out data structure, which is the queue, which is also implemented by a priority queue.We also have a blocking queue that helps us implement the producer consumer pattern very easily.

We then have the map, which allows us to store data as key value pairs, the most used one is the hash map.We also have a linked hash map and a tree map, which is a sorted map.

2) **What is the difference between an array list and a linked list?**

An array list uses an array for its internal implementation, whereas a linked list has

its own custom,note type with a previous and next pointers, it is like a doubly linked list.

Since an array list uses an array for adding an element randomly at any location, it can get very expensive all the way to big O(n),because we'll have to shift the current element in the array to insert an element at that location,whereas for Linked list it is very easy.

We simply create a new node and point the previous and next pointers appropriately.But when it comes to random access, arraylist is super quick because it is backed by an array, we can access any element by using the index, whereas accessing elements can get expensive in case of linked list will have to traverse through the doubly linked list.

So usually in applications that are very read intensive, array lists are the way to go.If we have a lot of read operations with a few right operations, then arraylists are the way to go. If not, you can use Linked lists.

3) **What is the difference between vector and classes like arraylist and linkedlist and hashtable versus hashmap?**

These hash table and vector classes are from previous versions of Java, where all the methods on them are synchronized, meaning if one thread accesses one of the methods on these objects, no other thread will be able to access them until the first thread finishes, which will degrade the performance of our application.

They are thread safe, but they can be very slow. That's the key difference. These classes are not synchronized.Their methods on these are not automatically synchronized.We have to take care of thread safety as per our application needs.

4) What is the difference between a regular hashmap and the linked hashmap,

The linked hash map maintains the order of elements in which we add them to the hash map, whereas the hash map doesn't do it.

5) How can we define our own generic class to create our own generic class

We create a class just like any other class right next to the class name.We specify a placeholder for the generic type within less than and greater than symbol.

This can be any alphabet, but by convention we use T for type once .We have that placeholder. We can use that placeholder to define fields, parameters to constructors, parameters to methods or even return types of a method

And to use this generic class, you create an instance of that generic class.And when you do that, you specify the type you want to use.

This is very specific.The generic type which will be replaced here, that T will be replaced with whatever type you want to use and you will pass in the appropriate data and use that generic type and you can use the same generic type with different data types or different types of data.

6) What is the difference between fail fast and fail safe iterators

Traditional collection classes like array lists, array set provide us the implementation of failed fast iterators.That is when we invoke the iterator method on these collections.The iterator that is returned back is a fail fast iterator.

If we use that iterator and loop through the elements, try to access those elements and at the same time, if we try to modify that collection, add, delete, etc on that collection, whether it is in a single threaded environment or a multithreaded environment, will get a concurrent modification exception fail fast.

Iterators will not allow parallel modification to happen, will get a concurrent modification exception.But when we use concurrent collection classes like copy on write Array list or copy on write array set,these classes provide us with the implementation of failsafe iterators.

They Return a Failsafe iterator, which will allow parallel modification, whether it is a single threaded environment or a multithreaded environment, will not have any issues iterating through and modifying the collection. At the same time, we see no exceptions.Those are called failsafe iterators.

7) Which collection class would you recommend to implement the producer consumer pattern?

Blocking Queue is the collection class that should be used to implement the producer consumer pattern,the blocking queue has a put method that can allow the producer to add work to the queue, and this method will block if there is too much work on the

queue and if the consumer has not already consumed a lot of work.

On the other hand, the consumer will use the take method that will block on the queue.If there is no more work to do, it will wait for the work to come in from the producer.So blocking queue is the collection class that will make it super easy to implement the producer and consumer pattern.

8) What is the difference between comparable and comparator interfaces?

The comparable interface provides a class the ability to define natural or default, ordering of its objects, comparable interface is from Java dot lang package.When a class implements this interface, it needs to provide the implementation for the compare to method that will be used to compare two objects of that class.

And this method's Logic should be in such a way that it returns a negative value. If object one has to come before object two a positive value object one has to come after object two and zero if both these objects are the same.

So we define our own logic and return these values depending on how we want the comparison
to happen.So Comparable provides the natural or the default ordering.And if we want to create any number of custom ordering for how the object should be compare, then we use the comparator interface, create our own comparator.

This comparator interfaces from Java.util package.And when we create a comparator class, we need to provide the implementation for the compare method from this interface, which takes two objects, compares them.

And even here we return negative value if object One has to come before object two positive object Two has to come before object one and zero if both are same.When we add our objects to collections, that is where these two are very powerful.

If you use collections like treeset treemap, etc. and when you add your objects to them.If you do not provide a comparator, then the default ordering or the natural ordering provided by the comparable interface. If your class implements the comparable interface and gives a comparative method, that ordering will be used.

And if you pass in a custom comparator you create, then that comparator will be used and to compare the compare method inside it will be invoked to figure out how your

objects should be sorted.

9) What are concurrent collections

What are concurrent collections when we use collection classes like array list. hash set, hash map in a multithreaded environment, when the first thread gets a lock on one of these objects,no other thread will be able to access the entire object unless the lock is released.

Also, these collection classes implement fail fast iterators.That is when we use one of the iterators written by these classes to iterate through the elements and also parallel try to modify that collection will get a concurrent modification exception.

That is where the concurrent collection classes like Copy on write array list, copy on write array set come in.And these classes allow multiple threads to access the same collection and modify it at the same time by creating a copy of that collection. And they will sync up the copy with the original collection later on.

And these collection classes, the concurrent collection classes will implement fail safe iterators, that is if we get that iterator from one of these collections.

Iterate through the elements, we can also modify the collection at the same time, all those problems with the traditional collection glasses are gone. In case of a map we have the concurrent hash map, which allows fine grained locking instead of locking the entire map. when there are multiple threads the locking happens at a very fine grained or a bucket level as
required.

Multithreading

1) What are the different ways in which a thread can be created?

There are two different ways in which we can create a thread.The first is where our class will extend the thread class and override the run method.The threading logic will go inside in this run method.

We then create an instance of our thread class and invoke th to Java, they are like anonymous functions or closure's where we don't have to use a function name, a

return type or even access modifiers likepublic, private, etc. What typically needs a lot of lines of code can be easily achieved through a lambda expression.
Lambda expression starts with parentheses, followed by an arrow symbol hyphen and greater than symbol, followed by the body of the lambda code.We want to execute within the lambda expression.

We can also pass parameters to the lambda function, just like how we pass parameters to a function.The advantages of using lambda functions are very few with very few lines of code.We can achieve a lot of things very easy to implement.

Anonymous inner classes wherever we want, anonymous inner classes.We can create them on the fly using lambda expressions and they can be even passed as parameters to other functions.

3) What are Functional Interface

What are functional interfaces if an interface has only one abstract method, then that interface is
called functional interface.

Examples of inbuilt functional interfaces are Runnable, which has only the run method comparator, which
has only one abstract method, which is compare to.

A functional interface can have any number of default methods that were introduced in Java eight, but
it can have only one abstract method.

We can mark a functional interface using @ functional interface annotation once we use this annotation
on an interface, if we try to add more than one abstract method to the interface, we'll see compile
time errors.

Functional interfaces can be expressed as lambda expressions.
That is a rule that the interface should follow to be expressed as a lambda expression. It has to be a functional interface.

If we want to express that interface as a lambda expression, it should have only one abstract method.

4) Can you explain to us with an example on how you have used lambdas to simplify coding applications?

One of the examples is when we spun off multiple threads within our application, we typically create a class that implements the runnable interface and provide a implementation for the run method within which the multithreaded logic will live, will then create an instance of this class when pass it to the threads constructor are we.And we invoke thread dot start.

It involves so much coding just to get a thread working.Instead, once you start using lambda expressions, it will become so simple.You will say Runnable are is equal to will start the lambda syntax the run method within the Runnable interface does not take any parameters.So we start with empty brackets.

Then the arrow hyphen greater than symbol than the body of the Function expression, lambda expression where the logic, formality threading will live, we can then simply pass this runnable instance to a thread and use thread dot start just like before.So all this creation of class will go away with lambda expression.

5) What is a predicate

A predicate is a function with a single argument that returns a boolean, true or false back. To create a predicate, we use the functional interface predicate that takes a generic type, and it has a single method called test that takes the generic type and returns a boolean value back.

Since it is a functional interface, we can express it using lambda expressions.Here is one example, we use the predicate passing the generic type integer, and on the right side we use a lambda expression to express this functional interface.

The implementation is very simple. It is the parameter to the lambda expression or the function, and the logic here is I is greater than 20.So that will return True, if I is greater than 20, but is less than 20 or equal to 20, it will return false.And here we invoke p dot test on that predicate.

We invoke the test method.This here will be used as the logic, as the implementation for

the test method.We get the appropriate result back predicate can work with different data types.

Here is an example of string predicate, which checks if the string length is greater than five, always a predicate returns a boolean value within the predicate.We can do whatever logic we want with the input, but it has to return a boolean value back.

6) What are Predicate Joins

Predicate joining allows us to use more than one predicate together, and we can also negate the result of a predicate here, have an example where a function takes a predicate as a
parameter and a integer array, and it applies this predicate on each of the array elements and it will print the array value only if the predicate returns True.

And up top here in the main method, we have an array defined and two predicates, one that checks if a number is greater than 10 and another one that checks if a number is even if we simply apply the first predicate which checks, if the number is greater than 10, then we will get all the numbers in this array which are greater than 10 on the console.No surprises.

And when we apply the second predicate, we'll get all the even numbers.No surprises there. But if we want to reverse this first predicate, if we want to get all the numbers that are less then 10 you can simply use P1 dot negate and we invoke that method. It will reverse the predicates logic or it will use false instead of true however you look at it.

So to the method here below, you will pass P one dot negate which will send you back the opposite results.

Instead of greater than 10, you will see less than 10 on the console.And if you want to apply both these predicates on the array, then you can use P1 and P2.That is, it will return all those numbers or it will display all those numbers on the console, which are both greater than ten and which are even if you use the end method on the predicate to accomplish that.

And finally, if you want or you simply use P1 or instead of P1 and so greater than 10

or even either of those.

7) What is the function?

Function is just like a predicate, except for it can return any type of value,not just a boolean value. We can pass in what type of data it will accept and the return type as well.

And we can express it as a lambda expression because it has only one method called apply it is a functional interface with a method called apply.

Whatever logic we implement using the lambda expression will be used and that value will be returned back instead of a boolean value.

8) What are default methods in interfaces and why do we need them?

When we create an interface and that interface is implemented by several classes in the application,all those classes will have to provide the implementations for the abstract method in that interface.

Which is fine, but in the future, if we change this interface and add more abstract methods, all those classes will have to provide a default implementation for those new abstract methods that
are added to the interface.Otherwise they will not compile.

All of those classes should be marked as abstract. That is where the default method that was introduced in Java eight comes in inside an interface.

When we define the new method, instead of making it an abstract method, we will define it as a default method.Default methods can have implementations.

It is where we'll put the default implementation so that the other classes that implement this interface, if they want to, can override this method.Otherwise, they will not.

But the code will not be broken, all those classes will still compile because we have the default implementation in the interface itself.So default interfaces were introduced.

Do not break the classes in the application when the interfaces change or new

abstract methods get added to the interface.

And if the other classes do not want to use those new methods.They can go with the default behavior.The classes, which aren't all right, can always provide their own implementation for these default methods.

9) Can a class implement two interfaces with the same default method A

class implements two interfaces which have the same exact default method

signature.

It is a conditional yes here I have two interfaces interface a an interface x which have the same exact default method default void m1 now if Class B implements those two interfaces.Let's see what happens.

Class B implements A comma X, we see a compilation problem immediately. And the error is that duplicate default methods named M1 with no parameters are inherited from type X and A.

We go towards the classical diamond problem to avoid this will have to provide an overridden version of this default method in this class B. Click here, say override, the default method and do provide your own implementation here and the error is gone.

So you can't just implement two interfaces and then stay calm. If you have the same default method in both of them, you'll have to provide an overridden version of that default method.

10) How can you filter out the even numbers in a given list using streams?

It is super simple to do that we take the given list, invoke the stream method on it, the first step to use streams is configuration.In this case, since we want to filter out, we will use the filter
method in the configuration step and pass it a predicate in this case.

The predicate is to check for even if it is even, it will return true.If not, it will return false.So the filter will filter out all the even numbers from the list and in the processing step we will collect all of them using that don't collect method and to the collect method we pass in a collectors dot to list that will give us back a new list that will have all the

filtered elements which are even.

11) Have you used any other methods on the stream other than just the filter method?

Yes, I have used filter dot count that will give us the count of the filter objects instead of collecting them to another list. We can simply get the count.

We can sort the given stream using the Sorted method.We can pass in a comparator to that sorted method which it will use.

We can easily create a competitor using a lambda expression and we pass that comparator to the sorted method in the configuration step and in the processing step we will collect the sorted list into a new list.

I have also used the max which will return the maximum element and we give it the comparator.It will compare and it will return back the maximum element in a given list.Similarly, the minimum. Will give me the minimum, the smallest element in a given list.

12) What is the difference between the filter and map method while working with streams?

Well, the filter method simply filters out based on a predicate we pass in the map method that takes the given list and converts it into a list with different content all together.

It will act on the given content and it will transform them based on the logic we pass into the map method using a lambda expression.

Java 9

1) Can we define private methods in interfaces?

Yes, starting Java 9 we can have private methods on interfaces, the reason or the advantage of it is to reuse the code across the default methods and the static methods

we can have an interface.
If we want to reuse some code across the default methods that we define, we define a regular private method, not a static one, and then invoke it inside the default methods.

But if you want to reuse the code across static methods, default methods, etc, then this private method also needs to be static.Only then they will be able to use it in other static methods on the interface.

2) How can we create unmodifiable or immutable collections in Java.

Prior to Java nine We have to use the collections utility class and the methods available on that class.But starting Java 9, we have the off method on every collection like list and set.

Have the off method which will give us back an immutable or unmodifiable collection when this method is invoked.

3) Stream API Updates

Three new methods are added to the streaming API on the stream, we can now use the take while method which will take the elements from the stream as long as the predicate that is passed into the take while method returns.True, it will stop as soon as the predicate fails or returns false.

It will not take the rest of the elements from the stream. Drop while is the opposite drop while will keep dropping the elements as long as the predicate returns True, and when it returns false, it will take all those elements.

And one more useful method is nullable on the stream which we can use to ignore the null values while using methods like flat map.

So take while drop while and of nullable are the three enhancements that were done to the streaming API in Java nine.

4) Enhancements to try with resource

Private the Resource Block was a feature that was introduced back in Java six, if we

define any resource within that try block, automatically that resource will be closed, provided that resource should implement the auto closeable interface and override that close method.

That resources close method will be automatically invoked for us when we use the try with the resource block in Java nine, it has been enhanced so that we don't have to define the resource inside the try block earlier we had to define it like this in here, but we no longer have to do that.
We can define or declare the resource itself outside and simply specify the variable name of that resource in the try block.

Java 10

1) What are some of the changes that happened when it came to the release starting Java 10.

Starting Java 10 Oracle promised a six month release before Java 10 up to Java 9. We had to wait for a long time between each release and each release Used to have a lot of features in them.

But starting Java 10 will master only a few features at a time, which will be very easy for the developer.

Java 10 introduces us to two such cool features using var to define variables and also some API updates in the Collectors' API.

2) What is the use of var that was introduced in Java 10. Var can be used to define inferred types.

Once we define a variable using a var the type of that variable will be inferred from the type of data or value we assign to that variable.No, it is not like JavaScript var in Java.

Once we assign a type of value to a variable defined by var we cannot assign a different type to it later on in the code that is possible in JavaScript but not in Java.

And also VAR is not a keyword. It was left out from the keyword list on purpose because we don't want to break someone else's code.

If someone had something like this in their code, for example, int var in earlier versions of Java, we don't want that to break their code.So for backward compatibility reasons var is not a keyword.

Hopefully in the future they might add it to the list, but for now it is not a keyword var is of great use when we define complex connections.So on the left hand side, we don't have to specify the entire collection type again, making it much more readable and we can also use it in loops, etc. When we are using complex collection classes, we cannot use VAR to define a lambda expression.

When we express lambda expressions on functional interfaces, we cannot use lambda expressions.It needs to have an explicit type and also we cannot define fields at a class level using the var keyword.We can use var's inside the lambda if we want to, but not on the lambda
expression itself.

3) Collectors API updates

While using the streaming API when we use the filter methods, etc., and create new collections from existing collections, how to create an unmodifiable list or a set as a result of the collect method. To do that in Java 10 we have the collectors dot to an unmodifiable list to a modifiable set to modifiable map, etc. These are the new api methods that are introduced in Java 10 we can use them and whatever comes back, we cannot change them.

It is unmodifiable if we try to change it, we will get an exception as follows.Will immediately see an unsupported operation, exception on immutable collections, so we get our immutable collection back when we use these to unmodifiable lists etc.

Java 11

1) Introduction

What are some of the updates that happened in Java 11, Java 11 adds new methods to the string API.

It also adds new apis to the file IO packages. it is added as an empty on the optional class, which is very helpful.And with every release, we'll have some depreciations and removals.

2) What are the new API methods that are added to the string class in Java 11

The first method is the blank method, which we can use to check for blanks within a given string, it will return.True, the complete string is blank.

We can have any number of blanks.It will return True In that case, if the string has some characters, it will return false.Next is the streaming update that happened, string dot lines method will take a string which has new lines, and it will return as a stream by splitting the string based on the new lines.

We can then collect, use the collect method on that stream and create a list out of it.Next is the update when it comes to spaces and unicode spaces, not just the regular spaces, when we use a string dot trim method, which is in earlier versions of Java, it will only trim the normal characters, not the unicode spaces.

So the Unicode spaces represented as follows backward slash u 2000 it is one way of representing a Unicode space.We can trim and clean up the unicode spaces .Using the strip method that was introduced in Java 11, Strip method will remove both the spaces at the start of the string and at the end of the string that is leading and trailing.

But if you want to be very specific, you have strip leading and strip trailing.And the last of the methods is the repeat method that will repeat a given string as many number of times as you want.

3) File API Updates

Java 11 also makes it easy to write text or string data to a file, we can use the update that happened on the files class in Java dot neo dot File package, file start.write.

String is a method that takes the path to the file and the content that we want to write to that file the string content and we'll write it to the file.

Similarly, files dot read string will take the path and return as the string data from the files so write string and read string make it super easy to work with strings while dealing with files.

4) isEmpty method

Java 11 also adds a is empty method on the optional class that is very useful when we work with reactive programming in prior versions of Java, we used to have optional dot is present where we can check if this optional has some value inside it.

But starting Java 11, we can also check if it is empty.Earlier we had to negate that is present to do this type of check.

But starting Java 11 is empty Method will let us know if the optional does not have a value for it doesn't it will return. True, if not, it will return false.

Java 12

1) String API Updates

Java 12 introduces indent and transform methods on the string class, the Indent method can be used for indenting the string that is adding spaces right at the beginning of the string.

It can provide positive value, which will add as many spaces as you provide.And if you do a negative value, if the string has spaces right at the beginning, they will be taken out.
The transform method takes a function and it will apply that function on the given string, the return type of that function need not be a string.It can return any other data type.So whatever that function returns, that will be the result of the transform.

2) Compact Number Format

What is the compact number format that is introduced in Java 12? starting Java 12 the number format class has a method called Get Compact Number instance, which, when used for formatting numbers, will express them in a shortcut format.thousand, for

example, will be represented as one k

So depending on the locale, it will use the appropriate shortcuts.And we can also pass in style like short, long etc while doing it instead of one m and one k.
If you pass in long, it will express it as one million, one thousand and

so on. **3) More Unicode Chars**

Java 12 adds support for several new Unicode characters to represent the characters in the Chinese language and also the support for representing all those characters on a chess board?

Here is an example.If I run this Java code, these are all the Unicode characters for representing some of the characters on the chess board like that.

You can represent everything on a chess board now using Unicode characters

in Java. **4) Collectors API updates**

Java 12 adds a teeing method to the collectors class, which takes two down streams and merges their results using the merger we provide.

So as we stream through a collection, we can use two down streams.We invoke the teeing method on the collectors and then you can use two down streams.

One downstream here is counting the total number of elements in the collection.The second downstream is filtering the results and storing them into a collection.

Both these results, the counting and this new collection will be stored using this merger class we provide.So the count will go into this variable and the filter new collection will go into the collection inside this merger class, that is how that teeing method works on the collectors.

Java 15 Features

1) What are sealed classes and interfaces

Sealed classes and interfaces is a preview feature that was introduced in Java 15 using which we can control which classes can implement a particular interface and

which classes can extend a particular class.

We do that using the sealed keyword.First, we seal the interface or the class and then we use the permitted keyword to specify those classes which can implement that interface or if it is a class that can extend that class.

2) Record Enhancements

Java 15, also enhances the record preview feature, we can use the sealed interfaces and classes with the record feature as well.

When we define a record, we can implement a sealed interface or extend a sealed class. Also, we can use custom annotations on the fields that are defined within a record.

When we define a record and define the fields for that record, we can use our own custom annotations
on those fields.

That is also a additional feature that was introduced in Java 15 record is still a preview feature in Java 15.

Spring Boot

1) What is dependency injection and what is a IOC inversion of control?

When we develop software applications, we organize our code across components and in the case of Java,these components are classes. **For example**, here we have a product Dao class on the right side, which is responsible for performing all the database operations on the left side.

We have a product controller, which is a Web layer class which uses the product Dao instead of we creating the object of this product Dao inside the product controller?

We delegate this responsibility to external frameworks or components like spring.Spring at runtime will dynamically create an object of this product DAO and inject it into the product
controller.

The product controller can then use this product Dao inside its methods. This process of External components are modules, creating the dependencies and injecting them into the required classes is known as dependency injection.

We as developers need not worry about how to create objects and all that, and we can focus on the business logic instead.

And the process of moving this control of object creation from the class to an external component or module like spring is called inversion of control.

It is a design pattern where we are giving away or inverting the object creation control from our application code to the external component like spring.
-
2) What are the Spring Bean Scopes

What are the two different spring bean scopes, they are Singleton and Prototype. Singleton is where only one instance of a bean will be created for the entire IOC container and the same instance will be injected wherever required.Singleton is the default scope.

If we configure the prototype scope for beans, then multiple instances of the beans will be created and injected wherever required.

Which one would you use and why?

If our application needs statelessness or if our application classes are stateless, we can go with the Singleton scope like controllers, DAO s, etc., which do not have any state within them.

And if they have a state, then we'll have to go with the prototype scope, because if we have state with Singleton, the data can be corrupted across multiple threads

But we don't have to worry about multiple threads in prototype because each thread

will get its own instance of a bean.

3) Can a prototype bean be injected into a singleton bean?

Yes, once the prototype bean is injected into the singleton bean at runtime, the same instance of the prototype bean will be used by the singleton bean.

4) What are HTTP Scopes

What are the different spring bean HTTP context, scope's, request session and global.

Request
scope is where a new bean instance will be created for every HTTP request coming in.

Session scope is where there will be only one bean instance used across the session that will have multiple HTTP requests and responses happening within a given session.For all those requests and responses, the same bean instance will be used.

The global or global session makes sense only if we are using Portlets within our application and this scope applies across portlets. It's like a global session which we can have across portlets and the same bean will be used across this global session or Portlets.

5) What are the Problems with traditional spring

Can you talk about the problems you used to face when you used a traditional spring framework before Spring boot was introduced?

When we use the various modules while working with spring framework like Spring core to do dependency injection,MVC to do web layered development and DAO, ORM to develop the data access layer, using hibernate, etc.,we have to use a lot of XML based configuration or Java based configuration to get the job done.

This can get cumbersome to maintain the application over time.And then we also had to make sure that we include all the modules or dependencies that are required for a project in the MAVEN pom dot xml or a gradle field file and ensure that these modules like the Core, the MVC, the Dao, ORM are compatible with each other as we move from one version to another.

And finally, once we did all this and developed the application, we had to manually deploy our application to an external web container like Tomcat or application servers like Web Logic, WebSphere, etc. So all these were problems before spring boot came in.

6) Why did you use spring boot in your project or what are the advantages of using Spring boot?

The first advantage or feature of Spring boot is auto configuration, that is we no longer need to write a lot of Xml or Java based configuration to configure Spring MVC to create a web layer or ORM using Hibernate Spring data jpa etc.

Once we add the required dependencies automatically, the dispatcher Servlet will be configured for our Web player.

A data source will be created for orm layer if we use the spring data jpa and also a transaction the manager is created and injected as well without us writing any code. So autoconfiguration is
the number one super cool feature of spring boot. Secondly, we don't have to worry about the module availability and version compatibility across these libraries or modules, which was a big headache with the traditional spring applications.

Thanks to spring boot starters provided by the Spring Development Team, these starters are ready to be used in our projects.

So if you are working on a web layer or a restful application, you simply add a Spring boot starter web.

Similarly, if you want to use the jpa spring data jpa if you want to use orm modules like Hibernate, you simply add that starter dependency to your spring boot application.

All the other libraries are dependencies that are required for the web layer for using jpa and hibernate will be transitively pulled thanks to these starters which hide all those complexities from us.

And also you no longer have to worry about the version compatibilities across these dependencies.

There is a parent project that every spring boot project extends and this parent project will have all that version information and the compatibility details for these libraries that are required.

So the module availability and the version compatibility problems are gone thanks to the starters and this parent project provided by Spring team Next, is the deployment, deploying our application once it is ready to production to any other container,were used to be a headache in case of traditional applications.

But Spring boot comes with an embedded servlet container when we launch our spring boot application.If it is a Web application is simply right click to run as spring boot application, that will launch it on an embedded tomcat by default.

You can also use embedded Jetti or undertow as your Web containers.So it's super easy to create and launch your applications using spring bootLast and very important spring boot also offer actuators which allow us to do health checks.

We can see the entire configuration, the auto configuration that is happening behind the scenes for our application using these actuators.

Check all the mappings that have happened internally for our application, the information, the health metrics, all that can be retrieved by simply adding one dependency to our Spring boot Pom or gradle build file.
We can do this even in production.So the health monitoring of our application can be easily done. So using Spring boot not only increases the speed at which we develop application, but the deployment as well as monitoring the application.
-

7) What is @SpringBootApplication

Can you talk about the Spring boot application annotation, the spring boot application annotation is the starting point of every spring boot application where all the magic happens.

It is a top level annotation, which contains three other important annotations. Inside it, they are the spring boot configuration enable auto configuration and component scan.

The Spring boot configuration annotation tells Spring boot that this class here can have bean definition's,we can define various methods that will expose out beans,

which will be later on used by Spring boot for dependency injection wherever required within our application.

That is what this Spring boot configuration annotation tells spring boot next enable.Auto configuration is a very important annotation, which tells Spring boot to enable and use auto configuration based on the dependencies we have on the class path.

For example, here in the Pom Dot XML, if we have a spring boot starter WEB Spring boot will automatically create a dispatcher Servlet bean and configure it for us to handle the incoming requests.

Similarly, if we have a spring boot starter JPA MySQL Connector, it will create a data source for us automatically by looking at the application dot properties and using the properties inside the application dot properties.

That is what this annotation tells spring boot to automatically create and configure those beans that are required for the application based on what we have on the class path

Last and very important, the components scan annotation tells Spring boot that it should scan through all the sub packages of the current package.

Wherever this class lives, which is marked with the components scan annotation indirectly through the Spring boot application, it should scan that package and all the sub packages for the spring beans that are marked with @ component at service, at repository, etc. and use them for dependency injection at runtime.

So Spring boot application does all that magically behind the scenes, thanks to the combination of these three annotations which are inside the Spring boot application annotation.

8) What is @SpringBootTest

What do you know about the spring boot test annotation. The spring boot test annotation uses a spring extension class to run our tests instead of using the typical Junit run.

It uses an extension class to run our tests. And this extension class knows how to search for a class within our application that is marked with @ spring boot application

annotation, use that class to create the entire spring context for our application that is

Why are all the beans load all the dependencies, do all that magic, and then that extension will start running our test marked with @ test J unit annotation.

So by the time our tests are run, all the spring context is created and the beans are available for testing.That is the magic of @ spring boot test annotation.

Spring Data JPA and Hibernate

1) What is Spring Data JPA and why did you use it in your project?

Spring data JPA makes super easy to create the data access layer for our application before spring data JPA to create the data access layer while working with ORM tools like Hibernate.

We had to create the DAO interfaces. The DAO implementations use entity managers or the Hibernate template,which in turn uses datasource.

We had to configure all, that to perform crud operations against the database from our application Spring Data JPA, says we no longer have to do all that boilerplate coding.We simply create a JPA entity class and a interface that extends one of the repository interfaces from spring data JPA API and they will be able to perform all the crud operations and much more on the database.

We no longer have to deal with entity manager factory and entity manager.They will be taken care of by spring data JPA,internally spring data JPA will use them.It is one more layer and it creates the data access layer.It also has support for finder methods where we don't have to write any queries.

And Internally, the queries will be generated against the database table by simply defining abstract methods in the interfaces.

It has inbuilt support for paging and sorting the results that come back.JPQL gives us the object oriented syntax to perform queries and we can also execute native queries where required.That is the reason we use spring data JPA within our projects.

2) How to use spring data JPA?

To use spring data JPA We had the spring data JPA starter dependency to the spring boot application,then we had the driver dependency.

If it is MYSQL we will add the MYSQL connector ,if it is Oracle we will add the Oracle driver dependency to your pom dot XML, we then have to create one class and interface depending on the application.

If we have one database table, we will simply define one model, class or entity that will represent that database table, we mark that entity with the JPA annotations like entity, ID, etc. and we then create a repository interface, not a DAO class, but the interface that extends one of the repository interfaces from spring data JPA.

And you tell it which entity it has to deal with and what is the type of primary key within that entity that set.

Once you do that, you will be able to use this repository interface in your controllers or any other layer where ever you want to perform the crud operations And if you want to define finder matters, this is how you can define the finder methods without writing any sequel and without writing any Java code.

You simply define an abstract method like this, and it will automatically generate the sequel required to fetch the data for you. And our application is run.

Spring boot will look at the dependencies that are there in your project.It will automatically create the data source by looking at the database dependency.

And the Spring data JPA dependency uses the properties in the application dot properties and it will create all the beans that are required and it will dynamically create an implementation for this interface using which you can perform various crud operations against your database.

3) Create Coupon Service Data Access Layer

We'll create a coupon micro service spring boot project and the data access layer for it using spring data JPA to do that, go to the spring tool suite file menu, new spring starter project.

Give it a name.Call it coupon service.The group name is com.bharath.Spring boot, the

artifact name is the same as the name we typed up top, the description is Coupon MicroService and the package.

You can copy the group name from up top and paste it right there.Click on next.
The first dependency you want to choose is the spring data JPA.So if you type in data JPA pick that
spring data JPA from under SQL, that is the first dependency, next type in MySQL and select the MySQL driver dependency.so that we can connect to the MySQL database from within our application finish that,will create the project.

Go to source main java, the first step is to create the model classes are the entities so you can right click on the package.

New class, the only entity we need is coupon entity that will go into a package called dot model or in some projects, they name this folder as dot entities instead of model finish.

The coupon class will represent the database table to become an entity we have an id code discount and expiry date.

So let's define all those fields, private long ID, private string code , a coupon code private. The discount I'm going to use big decimal to represent the discount from Java math, big decimal discount.And finally, private string exp underscore date

To keep things simple, we are using the varchar in the database exp date So ID code, discount and expiry date, we have all the fields defined getters and setters for these, you can go to source go to generate getters and setters do select all.

And I want it after the last field, which is after expiry date generate,control shift f to format.So far, this is just a plain Java class to make it a JPA entity.

You mark this class with @ entity annotation.This is mandatory and another annotation that is mandatory on an entity class is @ ID.

In this case, the ID Field is the primary key.So you mark it with an ID annotation.

And this ID field is an auto increment field if you see here in the database.So you use another annotation to tell that this is an auto increment field using @ generated value, annotation from java x dot persistence.

And the strategy that should be used will go in the brackets.Strategy is equal to generate then type generation type dot identity that will mark it as auto increment type.

So now this guy is an entity next you simply create a repository interface so right click on the package,new interface.

Give it a name, call it coupon repo and the package instead of Model use repos.This interface should extend one of the spring data JPA interfaces, this can be the crud repo that guy crud the repository are the JPA repository, which makes it even easier for us.
So extend the JPA repository.OK, finish.And here on this interface, let me Double-Click, to maximize it, we need to specify what type of the entity dealing with it is the coupon entity.

And the primary key field in this coupon entity is of type long, we need to tell it, what type is the primary key so long is the type thats it, so we have that data access layer ready and we can perform all the crud operations to test it real quick.

We have to configure that data source.So go to source main resources application properties and define the datasource properties, starting with the data source url so spring dot data source dot url is equal to the JDBC url which start with JDBC Colen MYSQLcolon two forward slashes.Local host is where MYSQL server is running 3306 is the port number slash coupon DB is the database name next spring dot data source start usernames.spring dot data source dot user name.

That guy, not that it's the user name.Username in my case is the root user, and then the last one is Spring Dot datasource dot password is equal to my password is test one, two, three, four.

Whatever your root password is, enter that right there.So that will configure their data source for us.And the last step is to write a simple test to test it.Go to source test Java.

Open up the test class, that would got generated, so the test class is marked with @ spring boot test annotation and this @ test annotation is a Junit annotation test.

Save coupon test, Save coupon is the method to test it.

We need to inject the repository, which is the coupon repo call it Repo Mark this with it @ auto wired annotation so that that will be injected when the application is run or when the test is run repo dot save is the method pass it a new coupon entity.

Select this new coupon hit control one extract to a local variable call it a Coupon set the field's on that coupon, coupon dot set.We don't have to set the id because it's the auto increment field, set the code within the double quotes

I'm going to call it supersale. Is the coupon code next coupon dot set. The discount is a big decimal.So you can use new big decimal pass in a integer value, say, a discount of twenty dollars.

Finally, coupon dot set the expiry date within the double quotes 12 12 2025.So that's our test, select the test, right click run as J unit test, as soon as we do this the entire spring boot application will be launched.

All the wiring happens and we should see a record getting inserted into the database. So we should have a record in the database. Go to MySQL Workbench do a select * from coupon.

Make sure you are using the coupon db to start with and then select * from coupon and you have the record in the database.

So it's that simple to create the entire data access layer in two simple steps. First,

you create the model objects, then you create the repository and you are ready to

go.

And if you want finder methods, you don't have to write any queries. In this case, I want a finder method to return back the coupon by code.

For that, you go to this coupon repo interface and simply implement an abstract method which returns a coupon object back find by B is capital y is small, which ever field you want to find by you, will enter that field name C capital.

I want to find a coupon by code and within brackets I pass in the string code that's it.You don't have to write any sequel's statements or use any annotations.

Now you can go to the test and write a new test,test find by code is the test that may add some lines at the end so that you can see it clearly at test annotation, Repo dot find by code and pass in the code

In this case, the code is super sale, that is what we have in the database.

And you can do an assert on this, let's hit control one assign statement to a new local variable. This is the coupon we are getting back take that coupon and do a assert, assert equals expected is the let's assert on the discount, the discount that comes back should be 20 and here the coupon dot get discount dot int value, that comparison will do it.

Now, select this test, just run only the test, find by a coupon test run as J unit test and see if the assertion is successful.

See that the test has passed, it is green, meaning we have got the coupon back by using the finder methods thanks to spring data JPAs finder methods, you don't have to write any query You simply use, find, by and then the field name as it is on the entity.

In this case, it is the code.You follow the camel case.That is why the C is capital here and you get the data back.

–

4) **What is an orm**

orm stands for object relational mapping, it is the process of mapping a Java class to a database table and its fields are members to the database table columns.

Once we do that mapping, we can synchronized our class objects into database rows. We as object oriented developers will deal with objects instead of writing SQL and invoke methods like save update, delete on the objects and automatically the sequel statements will be generated for us.

And using JDBC internally, these ORM tools will do the insertion updation deletion, etc. So we no longer need to deal with SQL directly.

And also we no longer have to deal with low level api’s like JDBC. we simply do the

mapping and then we start using methods like save and pass in our objects to it.

That is the power of ORM.

5) Create Product Service Data Access Layer

We will create the product micro service and the spring data JPA layer for that data access layer, to do that go to STS file menu, new spring starter project call it product service, the group name is com.bharath.Spring boot artifact is product service and the description is product micro service and the package is com.bharath.springboot , click on next pick the MySQL driver from up top and Spring Data JPA as well.

Hit finish that will create the product for us or the project for us.Go to source main java Right click start by creating the entity, call it product that will go into spring Boot dot model package, finish.

Product has four fields on it, the id,the name of the product,description and the price, let's add the fields for them, private long ID, private string, the name of the product.

Private string, the description of the product. Private big decimal.The price of the product.

Go to source generate getters and setters.Select all I want it, wanted after the last field which is price generate format it, save it. mark this with @ entity annotation and mark the id with @ id annotation also mark it with generated value, annotation generated value.

And the strategy we want to use is generation type dot identity. That's the id.That's the entity for us.

Next, create the repository right. Click new interface.
Interface name is Product Repo that will go in to repos package click on add. Select JPA repository that does spring data JPA repositories extent. Double click to maximize the generic type.

The entity type we are dealing with here is the product which we have just created and the id is of type long.We have the repository as well.

One last step is to configure the datasource in the application dot properties so you

can copy those properties from the application dot properties of the coupon service, control a control c, come back here, paste them and change at the end of it instead of coupon DB, change the database name to product DB.

product DB everything else remains the same, the username and the password for the testing, I'm going to leave it to you.

Go to Source Test java just like how we have tested the coupon service, you need to write a test here and save a product into the database by injecting the product repo which you have created injected into this test and write a test which will invoke the save method on the repo pass the product details of the product intended to do it and see if it gets there in the database.

6) What is JPA?

JPA stands for Java Persistence API, and it is a standard from Oracle to perform object relational mapping in Java ee applications.

Like any other standard from Oracle, JPA comes with a specification and an API, the specification is for the JPA vendors or the providers, the API is for us.

The developers are the programmers to master it.

There are several popular JPA providers like Hibernate.Open JPA, Eclipse Link, etc, which implement the JPA API by following the specification, the specification is nothing but a set of rules written in plain English so that these vendors can implement the API easily and consistently.

We as developers will learn one single API, so before JPA came into existence, we as developers had to learn Hibernate. API, Open JPA or Eclipse Link API, depending on which of these ORM tools we are using for our application.

But now we learn one single API and all these vendors or providers implement this API and we can switch from one vendor to another without making any changes to our application. That is the power of JPA.
Hibernate is the most popular JPA provider and used in most of the Java ee applications out there. So the JPA API is a set of classes that we master, will master entity manage factory entity manager or two or a couple of important classes in the JPA

API and a lot of annotations like Entity @ table, which we use to map our Java classes to the database tables and annotations like @ id @ column to map our fields to the database table columns.

This entity manager factory, an entity manager are specific to JPA.

Spring data will even hide this from us.We need not deal with this entity.Manager, factory and entity manager.

Spring data will make our life a lot more easier.So always remember that JPA is a standard for performing object.

Relational mapping in the Java ee world and hibernate, open JPA eclipse link are providers or implementers of this particular JPA, API or specification.

-

7) What are the different Entity Object States

Although you will not be directly dealing with these objects states, the entities are the domain class objects that we create, go into three different states internally when hibernate manages them at a low level, transient state, persistent state and detached state.

In a transient state, the object is not yet associated with the underlying hibernate session. It was just created and it is not yet persisted to persist.

It will be invoking either the save method on that object, on our repository, when we invoke save,it will go into a persistent state, or if that record is already there in the database, we invoke a find method and that record will be fetched and converted into an object which will be in a persistent state as well.

And when we invoke the delete operation, the object will go from persistent state to transient state again.

So it's not associated with the Hibernate session anymore. And that record is gone from the database.

But the object exists in our application and it is in a transient state.

Last but not the least the detached state, this is where the object was previously in a

persistent state, but if we invoke the session or entity manager, the low level api's, which we don't use in a spring data application or rarely used in a spring data application, and we invoke close and clear on them, the object will go into a detached state.
It still exists in the database, but now it's not associated with the underlying hibernators session or entity manager in JPA.

And again, if we invoke a save method on the detached object using another session or entity manager, it will return back to the persistent state.

The key here is the object being in a persistent state.

If it is in a detached state or a transient state, that object can be garbage collected and removed at some point because it's not being used internally in the application.

We can also pass the detached object across the application layers.So always remember that internally there is a transient state that is a persistent state.

When an object is associated with the underlying session or entity manager and a detached state to go into a detached state will have to invoke the underlying low level API like close clear on entity manager.

8) What are various JPA Associations

We will learn what JPA associations are or what JPA association mapping is? When we develop huge enterprise applications, let's say online shopping application, we don't just put all the data into one single database table like order.

We will usually normalize our database tables in to separate tables to hold the appropriate data like order, customer product address.

And each of these tables will have relationships with other tables, usually through primary keys and foreign keys.

That is where association mapping comes in to map this database tables to our domain classes or entities the JPA entities along with their associations.

We need JPA association mapping using that.

Once the JPA entities are marked with these associations, we can navigate from one entity to another.

Behind the scenes will have all the sequel queries Joins etc generated by the Hibernate or any other Jpa implementation.
JPA supports four types of relationships or associations, starting with one to one, the relationship between a person and a license is one to one.

A person has only one license and a license belongs to a particular person that is a one to one relationship.

The second type of relationship JPA supports is many to many.

An order can have multiple products and a product can belong to multiple orders as well. That is in many to many.

The last two are one to many and many to one, a customer can have multiple phone numbers, but a phone number always belongs to a particular customer.

So from the customer's perspective, it is a one to many relationship.But from the phone number perspective.

Many to one relationship, it's the reverse of it and also, these relationships can occur in two modes, we can configure them in two modes,unidirectional or bidirectional.Uni directional meaning only we'll be able to access that relationship in one order.

For example, in this customer case, if we configure one to many, only from customer to phone number will not be able to access the customer using the phone number entity. But we can access the phone number from the customer.That is unidirectional.

We can only configure it on either side of the relationship.But if it is bidirectional, that means we can access either of the objects or entities from either of them, we can navigate from customer to the phone number and phone number to the customer that is bidirectional.

So we can do all that association unidirectional bidirectional configuration using the four annotations provided by Jpa and several of its properties, starting with at

one to one.

At one to many, at many to one and at many to many will be using all these annotations and also various properties on them.

9) What is Cascading?

Cascading is the process of propagating the non select operations, like insert update, delete from the main object in the association to the child object, we can control how the propagation happens and at what level or what operations using the cascade element on the many to one one to many annotations.
It takes different values, Cascade is equal to persist means an insert operation on the main object should be propagated to the child.

Object cascade is equal to merge means an insert or an update operation on the main object should be propagated to the child object remove

As the property itself says, if you delete the main object automatically, the child objects should also be deleted.

And the next two will lead us to use the underlying entity manager, if you manually refresh.

A main object using the underlying entity, manager JPAs, entity manager automatically the child objects will be refreshed.

And detach will automatically detach the child objects if you detach the parent object manually using the entity manager, the child objects will be detached.

Last but not least, cascade is equal to all will work for all of these when a main object is saved, updated, Removed refreshed or detached the child objects will also be affected.

So all this can be applied to an association and depending on what you apply, it will control the cascading effect.

10) What is Lazy Loading

When objects are associated or related with other objects and when the parent

object is loaded, the child object can be loaded immediately, which is known as eager loading, or they can be fetched on an on demand basis, which is known as lazy loading.

For example, here we have a person and a person has an association with phone numbers that is a person. Person can have multiple phone numbers and the association is one to many.

If we define these phone numbers to be loaded lazily, if we configure it such a way, then when we fetch a person object from the database, only the person object will be loaded.

The phone numbers, the list of phone numbers will not be fetched from the database table. But if you invoke the get phone numbers method in the application at a later point in time, that is when all the phone numbers, records will be fetched from the phone number table that is lazy loading or on demand fetching.

This improves the performance of the application because we need not load everything unless it is required sometimes in the application, we might only need the person data. We might not even require the phone numbers.So lazy loading is where the data in the association is fetched on demand when it is first used and when the parent object is loaded, it will not be loaded right away.

So to enable lazy loading in our application on the annotations that we use, like at one to many, many to many, we have an attribute called Fetch and it takes two values.Fetch type is an enumeration which has two values eager which will fetch the data right away as soon as the parent is loaded.

The child will also be loaded and lazy, as you have already seen lazy.

Once you choose this, fetch time it will fetch the data only when the child data is accessed for the first time in the application.

-

11) What are two levels of caching

Hibernates supports caching at two levels, the level one cache is at the hibernate session level and level two cache is the session factory level, so session factory and session are the low level hibernate objects which are internally used by hibernate.

And when we didn't have JPA, we didn't have spring data,we used to directly use these in our applications, but now we no longer use them.

But internally, if we use level one cache, which is free, which is always there by default enabled, it happens at the session level.

But if we configure level two caching, which needs additional steps, which we will do, that will be at the session factory level.

A session factory, as the name itself says, is used to create multiple hibernate sessions. And so if we use level two caching, objects will be cached at the session factory level that is they are shared across hibernate sessions.

So if multiple users are accessing our application and we are using multiple hibernate sessions, the cached objects will be shared across those user sessions as well.

Level one cache comes for free, for example, when client one accesses our application internally, the hibernates session is used by JPA by spring data.

The very first time the data is loaded from the database and it's put into the cache.

The next time the client accesses the same data, the session will fetch it from the cache instead of going against the database.
And if another client access your application and if a different hibernates session is used, this cache will not be referred to, it will have its own cache, and the next time the client accesses it, this session will check against that particular cache and not the earlier cache.

So that's the difference between level one cache and level two cache, level two cache that will be a common cache across these sessions because it happens at the session factory level.So here is how the session factory level cache will look like.

A session factory is responsible for creating different hibernates sessions and it will have a common cache so all the sessions will share this cache so when this client access or application, the session loads executes select query loads the object and it will store it in the cache and then it will send it back to the client and then another client access our application, a different session is used.

But that session will first check to see if the data is there in the level two cache that is how we configure it.

Once we configure the level two cache, that check will be done and if the data is there, then there will be no database queries that data will be sent back to the client.

So level two cache is very powerful because data here is cached across sessions. But level two cache needs some additional work and hibernate does not have inbuilt support for it will use caching providers such as eh cache, which is very popular.

Server Cache, Jboss tree cache Os cache,Tangosal cache, etc., but eh cache is the most popular one and very easy to configure and very powerful, which will be configuring as level two cache for hibernate.

--

12) How to configure Second Level Cache

How did you configure a second level caching for your project?

The first step is to pick a second level caching library like Hibernate eh Cache or hazel cache Once we pick that, we add the dependency to the Pom dot xml,create a configuration file, for example, in case of eh cache will create a eh cache dot xml where we mention the temporary location where the cache objects should be stored, the maximum elements that should be stored in memory time to leave for those objects.

And all that information will be specified in this XML file.

We then go to the Spring boot project's application dot properties and turn on the flag, which is use second level cache to true.
And then we configure the region factory class from Hibernate, which is the eh cache region factory for eh cache Similarly, hazel cache will have its own class.

Then you point to the configuration file dot the eh cache dot xml, which is on the class path and after that configuration, you go to the entity class and you mark your entity classes which you want to cache using the @ cache annotation, and you can specify a caching strategy that you want to use.

-

AOP

1) What is AOP

AOP stands for aspect oriented programming.

This can be best understood by looking at an enterprise application which is typically divided into layers UI layer, business logic layer, Data access layer, and many more.

All these layers typically have some non-functional requirements like security, profiling, logging, transaction management, when we are especially using a database or a mq system.

Logging to write our errors or information to the log files, profiling to see how our application is performing and security, you know what it is for.

These are called crosscutting concerns because these are required across these layers as well as across the applications in our enterprise or even another enterprise.

Any application we develop will need all this and more.So that is where aspects come in and aspect oriented programming comes in.

In object oriented world, we define a class and object is the key unit that represents the class in the aspect oriented world and aspect is the key unit.

You can think of it as a specialized class that addresses one of these crosscutting concerns. It could be security, profiling, logging, transaction management, etc. So these aspects can be applied to our classes and objects at runtime.

Doing that will have several advantages.Number one, crosscutting concerns.

As I already said, we can address all the non-functional requirements which are common across the enterprise or even enterprises allowing us to reuse.
Once we develop an aspect, we can apply it across classes in our application and across applications.

Quick development, we can focus on our business logic without worrying about the

non-functional requirements,because we already have the aspects and we can apply them or we can create an aspect later on and we can apply it.

Focus on one aspect,this using this, we can specialize a particular developer interested in working on security aspect, he can develop aspect for that.

Another developer can work on logging, transaction management, etc., which will allow specialization.

Finally, a powerful feature is enabling and disabling aspects at runtime, we can enable them and we can disable them using configuration if we don't need them anymore.

There are several popular frameworks in open source Java world to implement aspect oriented programming,spring and aspectJ are two popular ones.

Spring also works together with aspectJ, and it has its own implementation of aspects as well. ---

2) What is the AOP Terminology

The four important concepts of aspect oriented programming, they are number one and aspect two an advise three point cut and finally join points.

An aspect is a plain Java class that can contain a number of advisers and point cuts. This class is where we address a particular crosscutting concern for our application. This could be security, transaction, management, logging, etc..

An aspect is made up of a number of advisers and advisers is a method that addresses a part of the concern. A join point is a point in the Java program where this advice needs to be applied. This could be a method, a field or a constructor.

So once we develop an aspect and a lot of methods in it, we need to apply those methods to the other classes and their methods in our Java application.

That is where join points come in. So these are the points where these advisers should be applied. This could be a method,a field or a constructor.

A point cut provides an expression language, it's like a regular expression language. To match a particular join point that is a field method or a constructor, it uses a syntax to express Join Points.

Let's take a look at all this while using spring and aspect J. When we create an aspect, we mark with @ aspect annotation from aspect J.

Within this class, we create a number of advisers to address a particular concern, which in this case is security for our application.

This apply security is a method in which will be writing all the logic to supply security for other classes and their methods in our application.So this is the advice.

We use the point cut annotation to express a join point using this expression, we are telling that this advice should be applied to all the methods that have a string.My method in them, in their name.

While using spring framework, the Join points could be before method after method after a method, returns after a method throws an exception and finally around, around works like before and after.It is a combination of before and after.

So if we use around, that advice will be applied both before a method gets called and after a method gets called spring supports method level join points.

There are no field or constructed level joined points in spring, so to summarize an aspect is a class that has a number of advisers, each aspect addresses a particular concern, like security transaction management for our application.

A join point is where a particular advice should be applied to the other classes and methods in our application and the point cut that provides expression language to express the join points. --

Transaction Management

1) What is a Transaction

A transaction is where everything happens or nothing happens.

For example, if we are working on a money transfer application, when the money is deducted from the first account and before it is credited to the second account, what if the application crashes, the money is lost?
It is neither in the second account, nor in the first account.That is where we will wrap such code in between a transaction or inside a transaction.

And once we do that, unless we commit, no permanent changes will happen.And if something goes wrong in between, we can always roll back the transaction to the previous state.

So that that money will still stay in the first account if something goes wrong.Transactions also allow us to have Save Points and roll back to that particular save point.

If we had multiple things happening in a transaction, multiple steps, then we can always roll back to the certain step instead of rolling back everything.

2) What are transaction ACID Properties

ACID properties stand for Atomicity consistency integrity and durability, and Atomicity is where it is atomic, meaning everything should happen or nothing should happen in case of a bank transfer, the money should be deducted and it should be credited.

Otherwise the whole thing should be rolled back.That is atomic or Atomicity consistency is where the data should always be in a consistent state that is in the case of this bank.

Transfer the sum total of the money in these accounts should be same before the transaction and after the transaction, it shouldn't be in a corrupted state.

Integrity is where one transaction should not affect the transaction when one transaction is happening, the data within the transaction is invisible to the other transaction until the commit happens the other transaction should not be able to see what is going on here.

Finally, durability. Once the transaction is committed, that should be permanent,

durable. We cannot roll it back once the commit happens.

3) What are Distributed Transactions

From this presentation, you will learn what distributed transactions are.

A distributor transaction also known as Global transaction or a xa transaction typically spans across different databases or even resources like messaging brokers.

A good example is money transfer between two different banks.We have to make sure that when money gets deducted from your one bank account, it also gets safely transferred to the other bank account or database.
So each bank has its own database and the transaction spans across databases.That is a distributed transaction.It could also be within your organization.

The distributor transaction need not always be across different organizations within the same company.

We might have applications which will be using multiple databases and we have to make sure that all of that work gets done within one single transaction.

A distribution transaction involves a transaction manager and one or more resource managers, the transaction manager is responsible for communication between our application and all the resource managers and resource managers know how to get the work done by the underlying resources, like the databases or messaging brokers.

They use a mechanism called Two-Faced Commit, so in case of distributed transactions, the entire transaction gets done in two phases in phase one, each participating resource manager.

Will be writing all the records to the local record, so the transaction manager makes sure that all the resource managers have written the data to a temporary location.

If there is any error, they will communicate that error or issue to the transaction manager. And if it is OK, they will communicate that status to the transaction manager in phase two.

The transaction manager now knows if all the resource managers are OK or not. So if they're all OK, then it will tell each of them to commit everything they have written to the temporary location.

If there is an issue, then it will ask them all to roll back to summarize.

A distributor transaction make sure that applications using multiple data sources can do it all or nothing.So everything should happen within the transaction boundary and they do it using transaction managers and resource managers in two phase commit.

4) What are the Transaction Isolation Levels

As the name itself says and if we have multiple transactions within an application which ran against the database, we need to isolate the underlying data for each transaction.

That is what we control using the different transaction isolation levels.So they give us transactional concurrency to run multiple transactions in parallel without affecting each other.

To understand the transaction isolation levels, you need to understand three database anomalies or transaction anomalies that exist, Number One Dirty Reads let's say transaction.
A update is updating the price information of a particular product, one two three.

In the meantime, even before transaction A commits, transaction B comes in and reads some data from the database with a query that qualifies what transaction is trying to do in this case, product number one, two, three.

Now Transaction B will get all the stale data because transaction A hasn't committed the data yet.This is called a Dirty Read at some point.

If transaction A rolls back the transaction transaction B would have already used the data which will leave the customer unhappy.

The price was less than thousand before second anomaly is non repeatable reads, this is where transection A read some data, in this case a product with ID one, two, three, transaction B comes in and updates the price two thousand now,transaction A

reads the data again, but it will see a different set of information as transaction B as committed the data.

So there is a difference between when the transaction A read the data for the first time and when it reads it again in the same business logic.

Finally, Phantom reads, This is where transaction A this is very similar to repeatable reads, except for the data qualifies in the where class.

In this case, transaction A is reading all the products that are priced more than 2000. Then Transaction B comes in and inserts a new product with price three thousand, which qualifies for this where class now transaction A reads the data again within the same logic for application or a piece of code, and now it will get a different set and additional product will come back in the results. This is called Phantom Reads.

Now that you understand these three isolation, these three anomalies at the database level or transactions,these anomalies can be avoided by using transaction isolation levels in the Java EE world, there are four transaction isolation levels, transaction transaction read uncommitted transaction committed repeatable read and serializable.

The transaction, read committed, if we use this transaction isolation level, the transaction can read uncommitted data.

This is where dirty reads will happen if we use this read, uncommitted level. Dirty reads non-repeatable reads and phantom reads can occur all of the anomalies will occur if we use the first level.

The second one transaction read Committed ensures that our transactions read only the committed data. So if we use this level, then we will avoid the dirty reads. Still will have that non-repeatable reads problem dirty reads are prevented. Non-repatable reads and phantom reads can still occur. Then comes the transaction repeatable read. This is the popular one most used one by default. Many databases have this as a default transaction repeatable Read.

This is where we can prevent the non-repeatable read problem as well.But the phantom reads can still occur.

Finally, we have serializable, which is the most strict transaction isolation level, and also the less performing as the transaction isolation level increases.

As we switch from Read committed, un committed to read committed then to repeatable Read and serializable. The performance of our application will go down because as we move closer to serializable, if we use serializable, that means it's almost like no two transactions can access the same set of data. At the same time, it could be a table level lock.

The different anomalies, basically dirty reads, non-repatable reads and phantom reads, and you also learn that these three can be avoided using their different isolation levels.

And serializable is the most strict and less performing isolation level that the three are OK to use based on the type of application you are developing.

Transactional, repeatable read is the most popular and most used transaction isolation level.

-

5) What is Optimistic vs Pessimistic Locking

What are optimistic and pessimistic locking mechanisms, both optimistic and pessimistic locking mechanisms, determine how the transactions in our application access the underlying data in the database.

Optimistic locking means a specific record in the database table is open for all the user sessions or transactions.

While using optimistic locking will not be locking any of the database records, instead, every time the transaction read some data, it will also read the version, number or timestamp.

These are two columns that we add to each and every table in our database and every transaction, and it reads the data will get these two.

And if another transaction updates the same record, it will update the version number and timestamp as well.
So the first transaction, when it is ready for an update, it will check if that version number and timestamp that it has initially read is the same in the database.

If another transaction has updated it, then it knows that the data is dirty and it has to

discard that transaction or re do the transaction by reading the data again.

So an optimistic locking. We are very optimistic and hopeful that another transaction will not update the data. So we are not going to lock the database records.

This is very helpful when our application is dealing with a lot of Reads and a few updates in pessimistic locking.

This is where our application will explicitly lock the records or record or even the entire table for read,right.Only for the current transaction.

The other transactions have to wait until the transaction finishes its work because we have exclusive lock on it, pessimistic locking provides better integrity than optimistic locking, but we have to design our applications carefully to avoid deadlock situations.

In pessimistic locking, appropriate transaction isolation levels need to be set so that the records can be locked at different levels.

The general isolation levels are read un committed, read, committed, repeatable read and finally serializable isolation.

It is not a great idea to read uncommitted data as the uses, it uses data from one transaction that's not completely committed from a different transaction.

On the other hand, serializable isolation is used to protect phantom reads. Phantom reads are not usually problematic, and this isolation level tends to perform, very poor, uncommitted are committed.

Read committed and repeatable reads are what we use frequently when we do pessimistic locking.

optimistic locking and pessimistic locking are two different ways in which we can control how transactions in our application access the data in the database and optimistic locking.

We have a version number or time stamp or both on each and every database record and every time a transaction reads the data, it gets them and before it updates the data, it will make sure that they both are the same.

If not, it will discard the transaction.
Pessimistic locking is where we explicitly lock the DB records or the table itself.

We have different isolation levels and read committed, and repeatable read are frequently used isolation levels.

Micro Services

1) What is a Monolithic Application

A monolithic application is where we put different modules in a software into a single application are code base, for example, if we are developing a hospital management software, which has a patient registration module that captures the patient details, the patient clinical module that handles the X-rays, blood tests and other diagnostic information bad management model to handle the inpatient beds and claim management to handle the insurance claims.

If we put all these modules into a single application or code base, it becomes a monolithic application or time as this code base grows, it will be very difficult to fix defects or to add new features.

If we fix a bug in one of these modules, we'll have to ensure that none of the other modules are impacted.

At the same time, if we want to add a new feature or fix that bug, do a build and do a deployment we will have to bring down the entire application affecting all the modules as well. --

2) What are Microservices

Micro services are small and focused applications that bring together pieces that change for the same reason.

For example, we can split the monolithic hospital management software into four

different micro services.

The patient registration micro service that gathers the patient details,the clinical micro service that handles their diagnostic information like X-rays, the bed management, micro service that takes care of patient beds, and the claim management micro service that does the insurance claims.
The code boundaries here are defined by the business, boundaries are the business problem these micro services solve, that is instead of having a huge code base for one single application.

These micro services will have tiny little code bases for the problem they are solving and they are autonomous, meaning they can be built and deployed on their own without impacting another micro service.

And if they have to communicate with each other to get the job done, they will do it using network calls through the APIs that the other micro services expose.

So if you have to define a micro service or if you have to check a particular application is micro service, you simply see if that application can be changed and deployed without changing or impacting another service.

3) Why Microservices?

Micro service have several cool characteristics, starting with heterogeneity, that is we can build different Micro service applications in different programming languages, depending on the business functionality and the requirement.

And we can deploy them to different operating systems as well.This is not possible in case of monolithic applications where we typically right the entire application in one language.

These micro services then can communicate with each other using restful apis messaging, etc., micro services are robust.

That is, even if one of the micro services goes down, the other micro services will continue delivering what they should deliver.

They are stateless, which enables easy scalability on Micro service applications, will not maintain state so they can be easily scaled when the load on the micro services increase, we can have different instances of the same Micro service deployed, which can handle the load.

Micro services are very easy to deploy, unlike monolithic applications where we have to deploy the entire application and all the modules in case are micro services, a change or a new feature can be easily deployed to production because we only have to deal with that particular micro service and all the other micro services will not be affected.

This increases that time to production, which is very key in applications today.Last reusable and replaceable.
We can use the same micro service wherever required.And since these are loosely Coupled, we can easily replace one micro service with another micro service if required some day.

4) REST vs Messaging

Should we use rest or messaging for communication between micro services, we can use both of them,but rest is usually good for Synchronous request response scenarios because it uses HTTP protocol,which is easy to implement, request response scenarios, and also to develop external facing API, which can be exposed outside of the organization.

On the other hand, messaging can be used for asynchronous communication, for non blocking type applications.

It is much more reliable because the messages can be persisted or stored and they will not be lost.And usually messaging is used for applications within the organization or for the micro services that communicate within the organization.
-

REST API

1) What is REST

What do you know about Rest?

Rest, which stands for Representational State Transfer, is a bunch of principles are architectural guidelines, http is the easy way to implement these guidelines.

Every application out there performs crud operations, create, read, update and delete. Which brings us to the first principle of rest, which is uniform interface and easy access using the http methods post, get, put, delete, etc..

We can perform all the crud operations through a uniform interface.Once we know these four methods, we can perform all the crud operations against any application out there.

We call these verbs because they do the work for us. Similarly, for the easy access principle of rest, http provides us uri s which can easily identify any resource. That is the reason they are called Nouns.
For example,to create a resource, we leave the HTTP post Method and the url to create employees, we leave employees slash employees url, and then the employee will be created and will get a http 201 response back along with the employee data.

Similarly, if you want to read a single employee back, you will use a http get method with the url which has the employee, ID , and they will get the employee details back with a http response 200 for update you will use the http put method passing the details you want to update and you will get the update details back, along with the HTTP response.

For a partial update, we use http patch method and pass only that data we want to update and we get the entire object that got updated back.

Last but not least for delete, we have http delete operation.Specify that id of the employee you want to delete and employee is gone.

So as a developer, once we master the http methods and the urls , we can do anything with any application.

That's the beauty of rest and https implementation of rest.

Rest should also support multiple formats, that's another key principle it should support xml to support Json.

It can support text or any other format, our applications once they say that they are restful, Web services are restful apis, they should be able to support multiple formats.

To summarize, the rest is a combination of principles are architectural guidelines mainly having a single interface.

HTTP is a great way of implementing rest because it provides us the http methods are verbs next easy to access.

We can use the urls in HTTP to easily access the apis that are exposed out and finally, it should support multiple representations that is different data types should be supported. -
2) HTTP PUT vs POST and PATCH

While creating restful APIs, when do you use the HTTP post and when do you use HTTP put?

Although there is nothing that stops us from using http put to create a resource in most cases, We'll be using an HTTP post for creating a resource and HTTP put for updation of resource.
What about put versus patch?

We use put method to update a complete object, whereas Patch is used to do partial updates. If you want to only update one or two fields on the object, then we use the HTTP patch method.

--
3) How did you create REST API?Can you briefly walk us through the steps that you follow to create a restful api using spring boot.

The first step is to create a spring boot project.

While doing that,we add the dependency, starting with the spring boot starter web dependency, which has all the restful api s and annotations that are required.

And then depending on what we are using for the data access layer, we add the appropriate spring data JPA or spring JDBC and driver jar dependencies.

You then create the data access layer using spring data JPA etc depending on what you are using,your code will vary for the data access layer.

But the key for the restful layer is to create a controller class, mark it with @ rest controller annotation.Map it to a particular uri,using the request mapping annotation at a class level,then we have annotations that we use as we create methods within this restful controller.

So bind these Java methods to the http methods using the appropriate annotation.If we want to bind it to post http methods, we use the post mapping if we want to bind it, to get use, to get mapping.

Similarly, we have annotations for HTTP, delete, put and so on.

And here url also provides a uri to which they should be binded to. when the client uses a HTTP post method.

With this url, This method will be invoked by spring web if you want to pass any path variables. This is how we do it.

We will have a placeholder in the uri mapping when we do it within angular brackets and we use @ path variable annotation provided a placeholder name which we have used up top, and that variable will be taken out of the url.

That value will be injected into this parameter and we can use that inside the method.
Create an application with the right dependencies, mark the controller class with the rest controller, then request mapping to that controller,a path url and create as many API methods as you want.

Bind them to the HTTP methods using the appropriate annotations

and path. -

4) Create Coupon Service REST API

The restful API from the coupon service that will allow other applications or clients to create coupons and also retrieve the coupon, given a coupon code,to do that, go to the Pom Dot xml of the coupon service.

Start by adding a new spring starter dependency, you can copy the spring data JPA dependency paste it above and change it from data hyphen JPA to Web.

So it is spring boot starter web, this guy has everything we need to write our restful

controllers. Go to one of the packages.

Right click create a new class called Coupon Rest Controller and this will go into a package called controllers.Finish.

The first step to make this a restful controller is to mark it with @ rest controller annotation, and we can map it to a particular uri.

So request mapping is the annotation that we will use.

And within the double quotes forward slash API is the uri. If we want to map to this particular controller.This is going to have two methods.

The first method will save the coupon into the database and it will return the coupon details back to the public coupon.

This will save the coupon, will call it create coupon, create coupon, it will receive a coupon object and it will return back coupon object, hit control one import coupon because it is in the model package,hit control one add a written statement that will return the same coupon for now will change that in a second.

Next, mark this with @ Post mapping annotation at post mapping because http post is the method that will be used to create coupons in the database.
We are going to add a coupon to the existing collection of coupons, within the double quotes slash coupons is the url.

Within this method, we need access to the coupon repository, so auto wire that coupon repo is the interface, call it repo, mark it with @auto wired annotation.

Inside the Method you use Repo dot save, pass in the entity that came in the request coupon. And we can simply return back what the save method returns, the save method will always return the coupon that is saved to the database, which will have the ID on it, the auto increment field. So we will return back the coupon that just got created in the database.

And this coupon,when we receive it in the request, we need to mark it with @ request body annotation, so that it will be serialized or deserialized that the Json that comes in.Will be deserialized into this coupon object and the next method is to find the coupon by code public,it returns coupon as well.

Get coupon by code is the method name.This guy receives a string coupon code and he should return a coupon back. So Repo dot find by code we have that method, which we have returned earlier.That is what should be returned back.

Whatever that matter returns that will go back, mark this method with @ get mapping. So http get is what we use to retrieve coupons, so we use http to get mapping annotation within the double quotes that url.

All that you want to use is slash coupons,slash.This is the placeholder.

So it goes in angular brackets code whatever code comes in, in the url that will be automatically put into this.

And you use an annotation call @ path variable, @ path variable within the double quotes you specify the name you have used here.

That is how we bind it.So when we pass in coupon code like Super sale Spring will take it,Spring boot will take it, and it will automatically inject it into this parameter, we'll use it on our method, switch the coupon and return it back.

Let's add a context and a port Number to do that, go to the application dot properties. And here, if you type in context, hyphen path, use that server dot servlet dot context path. That is how we define a context path for our application. Coupon service is the context path we want.

And then the Server dot port number is equal to 9090 for the coupon service and for the product service.
We will use 9091 later on. Run this application. Right Click Run as Spring boot app. Save everything.

That will launch the application on Port 9090,we have one problem, which is in the application dot properties, which says the context should always start with a forward slash go to the application dot properties right there where we define the context path there should be a forward slash. That's it.

Run the application again, go to the play button, start the coupon service. And that should launch the application on port 9090, and we should be able to retrieve the coupons that are there already in the database or we should be able to create new coupons.Go to Postman. that will be our testing tool.

Open up a new tab, localhost : 9090 is the port number slash coupon service slash API because we have mapped our coupon rest controller to slash API, slash coupons, slash code slash coupons, slash the code we have in the database is super sale.That is the coupon we have in the database.

Let's try to retrieve it, localhost 9090 coupon service ,we have to assess there it should be a coupon service.That is why we get a 404 and send it again. This time we get the coupon back.

Let's try to create a coupon, copy this data, open up a new tab, change the method to post. Go to Body select raw.

Change this type from text to Json, that is the input we want to send and paste the data we have copied. Let scroll down.

We want this coupon code to be mega sale, take out the ID field, and we want to give a fifty dollar discount and the expiry date.

We are going to leave it as it is.Go grab that URL from the previous window, come back, paste it.The post url, these slash coupon we are adding a coupon to existing collection of coupons. That is how this url should be used.Hit send.

That creates a new coupon in the database with ID two and we get the

details back. So both our rest ful end points will work as expected.

This restful end point will be hit by the product Micro service to fetch the discount. -

5) Create Product Service REST API

We will create the restful API for the product service that will allow the client applications to create a product to do that, first go to the Pom dot xml, go to the spring data JPA dependency, grab it, copy it, paste it right above, change it from starter data JPA to web.

Do a control shift f to format. Save it right click on one of the packages, create a new controller class, call it product rest controller that will go into a package. Spring boot dot Controllers finish.

Mark this with @ rest controller annotation, next, also mark it with @requestmapping annotation, to map it to a uri slash API is the url that we want to use.

This guy will have only one method that will return back a product once it is created, so create product is the method name that takes a product model, object calls it product and it will also return back a product, hit control one and add a written statement.

Mark this with @ request body so that it will be deserialized.When the request comes in mark this method with @ post mapping, because we are creating a product or adding a product to existing collection of products, or if there are no products, we'll start from the very first product. So we create a product within this method, so we have to inject the repository at this point, product repo is the interface that we have created earlier call it repo mark it with @ auto wired annotation, return, repo dot save and pass in the product that came in the request.

But before the save happens, we want this restful controller to make a call into the coupon service and apply a discount on this product,product's price before it is save it to the database that is where the rest template from Spring web will come in and we will use that to make restful calls to the coupon service.

---

6) Use RestTemplate

We will make a restful call using the rest template from the product service to the coupon service. To do that, inject a rest template. Call it a rest template and mark it with @ auto wired annotation.

We need to expose a bean of this type, automatically it will not be created for us.So we go to the application class that would have been generated.

When we created the project and will define a new method java based configuration, the public rest template is what we want to return back. Call it anything. Call it a rest template.

And within this, we return a new rest template, object out and mark this with @Bean annotations so that spring will create a bean and use it to auto inject it. Or auto wired into this product rest controller.Now we use this rest template, make a restful call to the coupon service.

So use rest template dot.It has several methods get for Object get for entity post put and so on.

So depending on the http method that you want to use, there is an appropriate method.Right now we want to use a http get to make a http get call.

So get an object, pass in the url in the double quotes and then the response type. So we are expecting a coupon to come back.

But this product service does not have a coupon class to handle the response that comes back. So go to the coupon service, go to the model, copy this dto.

This is a model object coupon dot java,come back.Create a package

here. Right Click on the source main Java End product service.Create a

new package.

Call it com dot bharath dot Springboot dot dto, a data transfer object.This will be a simple Pojoaque paste the coupon on it.

Open the coupon dot Java double click to maximize get rid of all this JPA annotations, it

is not a model object, it is simply a pojo that we are going to use,to handle the response so you can get rid of all these jpa imports. Save it.

Go back to the product rest controller.The response type is coupon dot class.

Hit control one on this line, assign a statement to a new local variable and name it coupon. So when we hit the restful url, which we are yet to configure, we'll get a response of type coupon.

This call is just like what you have done earlier from Postman, this guy with the coupon in the url now we need access to this url so copy this url go to your method.

We don't want to hard code this url right here.So we will configure it in the application dot properties.

Go to source main resources application dot properties.We call this coupon service url. So coupon service dot url or coupon service url however name it or let's call it coupon service dot url is equal to paste that url the coupon code itself will not be hardcoded that will come in the request when the request comes in to create a product, that coupon code will be sent in so add HTTP at the beginning of it, HTTP two forward slashes localhost 9090 coupon service api coupons.
Now go back to the product rest controller.Inject that value, the constant that we want to use.

Call it string, coupon service url mark it with @ value annotation from spring within brackets, within the double quotes use the angular brackets outside of angular brackets a dollar symbol so that it will pick it from the application dot properties and injected go to application dot properties, grab the name of the property we have configured come back paste it within those braces and that will be injected for us we use that coupon service url here.

we need the code.So the code information we apply that discount for this product will come in when this product request comes in.

If we go to this entity, this product entity does not have a coupon code, but we'll add it as a private string coupon code.

This field doesn't exist in the database and we don't want to save the coupon code in the database. We only need it in the restful controller when the request

comes in to apply that discount. So mark this with @ transient annotation.

Any time we don't want a particular entities field to be mapped to the database, you mark it with @transient annotation from java x dot persistence, hit control one, create getters and setters for it.OK, save it.

Now go back to the controller.Here we will append plus to this url product, dot get coupon code once we have the coupon code it just like this url 9090 at the end of it.

It Will have a coupon code that needs to be sent to the coupon service. We get the entire coupon object back before it is saved.The product is saved to the database.

We say product dot set, the price of the product to product dot get, the current price dot subtract.The big decimal has a method called subtract.Subtract coupon dot get discount. So we are subtracting the discount.

That this coupon applies to a particular product from the current price of the product and then saving that product to the database.
--

7) Test End To End

We will test the product rest API before we go ahead and do that, go to the application dot properties.

It should be a coupon service dot url once we come back to the controller here as well in the value and we do the injection.
It should be a coupon service dot url next we configure a context path for the product web application type in context hyphen path is equal to forward slash product service is the context path.Next server dot port is equal to 9091 is the port number we want to use for the product service.

Let's try to bring our application up. Right click Run as Spring boot application.Go to Postman, go to the post request, which you have sent earlier, to create a coupon,copy that come to a new tab, method is post, go to the body of it, select RAW, change the type from text to Json, paste what you have copied.

Go grab the post url from earlier.Come back paste it here, change the port

to 9091. Change Coupon Service to product service. API is fine.

We are adding a product to an existing collection of product slash products and the information we need to send in is the product name, say MacBook Pro.

The price of the product and the description of the product, all that needs to go in, let's go check the model object quickly to pass in the right fields in go to model, or you can go check the database as well.

We have four fields out of which we need to pass in name, description, price and the coupon code, name,description.

Within double quotes, its price is a numeric value, say two thousand dollars comma. The last one is the coupon code : coupon code we want to use is a super sale that we have in that database when we pass the product.

Service should use this coupon code, make a restful call to the coupon service that is already up and running on port 9090

Apply that discount to these two thousand and then save it.This product information in the database.

If all that happens by hitting send.Lets make sure the url is right localhost 9091 it should be product service,slash api slash products, send, see the discount of twenty dollars applied on it.

So it has saved the MacBook Pro to the database.we can go check our database, go to the product database, use product DB, execute that to a select *r from product, execute that and we see a new product, MacBook Pro,1980 is the price all they have and in the price as 2000, the discount has been applied because the super sale has a discount of twenty dollars.
So we can play around with other coupons as well.Try to create a product, apply a different coupon and that price should be or the discount should be discounted from the actual price.

8)What are Spring Boot Profiles? How did you manage the application configuration

information like the JDBC urls, the username password, the restful urls and so on for your application?

We have used spring boot profiles that will define the configuration information for each environment as it changes from dev to testing to stage to prod, will create a separate application dot properties file and we'll name them as follows.

Application hyphen Dev for Development Application hyphen Test for testing environment application. Application hyphen prod for properties.

These test prod Dev that we use in the file names are nothing but profiles.To activate a particular profile.

We use the following property in the main application dot property Spring boot Profile dot Active. Once you specify whether it is Dev Test or Prod, automatically that configuration will be picked up and used in that environment.

Another recommended way is to use the JVM argument as follows.

When the Spring boot JAR file is launched, during the Java command we pass an hyphen d spring dot profile dot active and the profile that should be activated and used and automatically Spring boot will use the appropriate file for configuration.

So Spring boot profiles are the answer for our configuration across environments as the urls user names etc change.

SOAP Web services

1) What is SOAP

What is soap? SOAP is a Web services standard, which stands for Simple Object Access Protocol.

It allows two applications or multiple applications which are built in different programming languages,running on different operating systems to communicate with each other using the http protocol and xml.

It comes with some predefined XML elements that we will use to exchange

messages across these applications.

-

2) What are the Java EE Web Service Standards? What are the Web service standards in Java?

We have two service standards, one for the soap based Web services and one for the rest based Web services, for the soap web services the Java ee standard is Jax ws., which stands for Java API, for xml based Web services.For the rest Web services.It is the JAX RS, which stands for Java API for xml based rest services.

Although it says XML, it supports various data formats in rest, like any other Java EE specification or standard.It comes with that specification and that API.

The specification is for the Web service providers or frameworks like Apache, cxf , Apache axis in case of soap and for the RESTful World it is Apache cxf again.

And then we have frameworks like Jersey.

These frameworks implement these web service standards and this API is for us, the developers.

Once we learn the API, we can use any of these frameworks. Our code and our application will not change.

The same API will work across these frameworks that implement these

standards. 3)What are the Two Types of SOAP Design

What are the two different ways in which we can implement soap based Web services?

We can use the wsdl first or top down approach where we create the wsdl file first and then generate the Stubs and implement the web services endpoint.

or we can go the Java first route,which is also known as bottom up Web services, where we create the Java classes first, mark them with the appropriate Jax Ws annotations, and then we can generate the wsdl file from that class.

We create our interfaces, we create that and is called the Java Plus.So we

have two approaches of wsdl first and Java first.
-
3) What is WSDL? What is a wsdl file?

WSDL stands for Web services description language, it is an xml based file which acts as a contract between the social services provider and the Web services client.

Once we create a wsdl file,we can use it to generate the Java or any language stubs and implement the soap web service on the provider side.

And also the client can use this wsdl file to generate the stubs and to create a client code to invoke those Web services as well.
--

4) What is the WSDL Structure.Can you briefly explain the structure of a wsdl file?

The root element of a wsdl file is the wsdl definition's element within which we have the abstract section and the physical section.

The first element of the abstract section is the wsdl types.

This is where we define all the schema types that we will use later on to define the messages, the input and output messages to our Web services, operations or methods.

Once we have those messages, which we create using the wsdl message element, we use those messages to create the operations.These are like the methods and these methods receive inputs and send outputs.

So we'll be using messages in here, all the Web services operations for a particular Web service end point will go inside the port type wsdl element.

Next is the wsdl binding.

This is where we specify what type of binding we want to use on the request and response message types.

Then we move on to the physical side of things using the wsdl service and wsdl port we defined on which url,this website which can be accessed.

The connection between the abstract portion and the physical portion of the wsdl is this binding element on the service when we define the port. We use this binding which binds it to this binding up top here to the abstract section.

-

5) What is the Top Down approach

What are the steps to implement a top down or contract first or wsdl first Web

service? We start by creating a wsdl file based on the business requirements.

Once we have a wsdl file with all the operations, the bindings and all that, we will generate the stubs, the Java stubs from this wsdl file using the code gen plugin in case of a framework like Apache cxf, it has a cxf code gen plug in.

When we point this plug in to the wsdl file, it will generate the Java stubs for us once we have the generated Java stubs will create a class that implements one of these Stubs.

And will Override the method inside it that receives the soap request and returns a response back and we mark these classes with the appropriate annotations, like @ Web series, annotation from Jax WS java ee Standard.

6) What is the Bottom Up Design?

How do you implement a Java first or Bottom-Up Web service?

To implement a Java first Web service, we start by creating an interface where we will define all the methods we want on that particular Web service will mark this interface with the @ Web service annotation from Jax WS and then we mark these methods with @ web method annotation.

The parameters of this method will be marked with @ Param and the result will be marked with @ web result from Jax WS.

Once we have this interface, we implement this interface by creating a implementation class and we implement all the methods in that interface at runtime frameworks like Apache, cxf will scan through these classes,Annotations that we have used and the information we have provided, and they will generate the wsdl file

on the fly, which the clients can use to create client applications.
-
7) What is a SOAP Client

How do you create a soap web service client application to create a soap service client application?

We need to have the wsdl file that the Web service provider uses.
Once we have that wsdl file, we copy it to our project, generate the Java stubs from the wsdl file using plugins like apache cxf code gen plugin.

Once we have the generator stubs, it is super simple to invoke the Web service.

We create a client class and within the client class we'll use the generated service and port type classes pointed to the web service url passing the appropriate request and we get the response back.

All the serialization and deserialization will be taken care of by frameworks like Apache CxF. -
8) What is MTOME

How to send attachments or upload and download files by working with social web services?

MTOME which stands for message transmission and optimization protocol, is the best way to send attachments to enable.

MTOME while working with Apache cxf we can enable it at the bus level or we can enable it at a port binding level for a particular web service once we turn on MTOME It is super simple to use attachments in our implementation classes.

For upload, we will use this data handler from java X activation as the input parameter. This will have the file, the uploaded file data will come into this automatically at runtime.

We can read it and we can do whatever we want with that file.Similarly, if we want to send a file back, you simply return back a data handler and file source.

9)SOAP vs REST

What are the differences between rest and soap web services and which one should be preferred?

Rest Web services are very simple to implement and use,they perform well when it comes to the messages themselves.

The messages are very lightweight.They are stateless.So the applications can be easily scaled as the load increases.

And it supports multiple data formats, not just XML, which is the case.

Soap based Web services soap used to dominate the Web services world before rest. But it is slowly becoming legacy because it is a little hard to implement. Soap Web services compared to rest.

It has the message overload where we have to use some predefined soap headers, the soap envelope and all that when it comes to messaging and it only supports the XML data format.

Soap is still being used where we need a very strict wsdl contract where the non-functional requirements like encryption, decryption at the message level and also reliability of the messages is required right out of the box, several soap implementations like Apache's cxf etc. have all these features in built and that is one use case I see where soap is still being used.

So soap is still being used for internal applications within the organization where these requirements are of high value.

Otherwise rest is the preferred way to go. Most of the public facing APIs today are restful Web services.
–

Security

1)What are the Components of Spring Security

What is the spring security flow and the important components that are involved when the request comes in from the Web browser or from a restful client to a secure spring boot application?

The first component that intercepts this request is the authentication filter.It is a servlet filter that checks if the request is authenticated already, if the user or the client is already authenticated.

If not, it forwards to the authentication manager, the authentication manager component intern uses a authentication provider.

This is where the authentication logic lose authentication provider.

In turn, uses a user detailed service to fetch the user details from a database or LDAP or any other source.

It also uses the password encoder to encode the incoming password because we don't deal with plain text passwords in spring boot.

If the authentication is successful, the appropriate response will be sent back.
If the authentication fails,the authentication provider will hand over the response back to the authentication manager.

The authentication manager hands it back to the authentication filter, the authentication filter based on the response.

If it is successful, it will use the authentication success handler and stores all the user details and authentication details in a security context object.

If the authentication fails, then the authentication filter will use the authentication failure handler to send the appropriate error response back to the client.
–

2) How did you secure your REST APIs?How did you secure your restful micro service applications?

We have used Oauth along with JWT to secure our micro service restful

applications. 3) What is OAuth?

Oauth stands for open authentication and authorization standard, it allows one application to access the user's data within other applications without the user directly sharing the user id and password of these applications with the application that needs access to this data. –

4) What are the Key Components in OAuth

What are the important Oauth architectural components?

There are four components in Oauth workflow.The first one is the authorization server. This is the guy that will authenticate and provide or generate that token.

Second is the resource server.This is where our secure resources will be stored our applications live.

Third is the resource owner, the guy who has the ownership on the resource and who grants permissions to those resources.

And finally, the client who needs access to those resources on the resource server. -
5) What is the OAuth Workflow?Can you explain the Oauth workflow?

There are different workflows possible in Oauth depending on the use case, but a typical workflow will start when the user tries to access a client application.

And this client application needs access to other resources that this user owns.

At this point, the client application will ask the user to share the user id and password with the client application itself or it can ask the user, the resource owner, to share the user id password with the authorization server.

So the authorization server, once it gets the user data and password validates, it sends a token back, the token is generated and the token is shared with the client application.

The client application now can use this token to communicate with the resource server and the resource server will validate that, this token is a valid token by communicating with the authorization server at that point if the token is valid.

The resource will send the data this client application requires.

-

6) What are the OAuth Grant Types

Depending on the use case, we are working on the workflow using which the username and password will be exchanged between the end user or the resource owner and the authorization server and the client application can differ.

Grant type allows us to configure these different workflows.

-

7) What are the Different Grant Types

Can you explain the different Oauth grant types?

There are four different grant types, starting with the authorization code.

When we configure this grant type, the client application will redirect the end user or the resource owner to the authorization server will have to authenticate with authorization server. Once that authentication is successful, the authorization server will tell the client app that the authentication is successful and then the client app will request for a token, gets the token, uses it to communicate with the resource server to get the required resources.

Next is the password grant type when we use this,the workflow is that the client app will directly take the user name and password from the resource owner or the end user uses them to communicate with the Oauth server to get a token and uses that token to communicate with the resource server, third one is the client credentials,here there is no end user or the resource owner.

This workflow is typically used for communication between micro service applications.The client app will have a user id and password given by the authorization server. It will store those details locally and it will use them to get a token when required

to communicate with the resource.

Server this is very useful in case of single sign on applications.Next is the refresh token. The Oauth token will expire at some point.The Oauth token that is issued when we use any of these grand types, it will expire after some time if it is not used or after that time finishes.

So we can use if we configure the refresh token on grand type, that client app can refresh and get a new token Instead of going through the entire workflow of these grand tapes, it can simply use the old token and get that new token from the authorization.
-

8) What is JWT in a typical Oauth workflow?

When the client application sends the Oauth token that was given to it by authorisations over to the resource server. Resource server has to make sure that the token is a valid token for that.

It will communicate again with the authorization server to check if the token is valid. Secondly, the token that the client application sends will not have the user and user roll information which are required for the resource server to check the access level for that.

Again,the resource server will get all those details from the authorization server that is where the JWT comes in.

JWT stands for Json Web token and this JWT will have all the user and user role details when the token comes to the resource server, it need not communicate with the authorization server to get those details that token will already have them.

And secondly, JWT tokens will be signed by the authorization server.So the resource server when it receives the token from the client, it can verify that signature using the public and private keys and it may not make another call for the validation.

If this signature is valid, then the token is valid.
9) Hot to configure JWT

What are the steps to configure and use the JWT while using Oauth in spring security?

The first step is to go to the authorization server project and in the authorization config will start using the JWT access token converter to create JWT tokens instead of the regular tokens.

The Oauth tokens will use the JWT access token converter and generate the JWT tokens and will store them using the JWT token store while creating these tokens.

There are two ways in which these tokens can be signed.

One is the symmetric way where we use a private key or a symmetric key that will be shared with the resource server project as well.

So both the authorization server and the resource server will use the same key to check, to sign and to check the authorization server will sign the token using this key.

The resource server we'll use this key to symmetrically validate that the token is valid and it is signed by the authorization server we can also use asymmetric key where we use the Java key tool to generate a key pair, a private key and a public key.

The private key will be used by the authorization server, will store it in a key file, will put that under our source and resources of the authorization server.

This jks file, which is generated using the Java key tool that comes with JDK installation. Once we have the key file, it will have both the private and public keys.We'll share that with the resource server as well.

The resource server will use the public key, which is inside this jks file to verify that token when it receives the token and the authorization server signs the token using the private key, which is in that dot jks file.
--

10) How to rotate tokens? How often did you rotate tokens in your Oauth application depends on the use case?

If it is a read right application, like a financial application or a banking application, we will rotate them every five and 10 minutes.

If it is more a read only operations like a Linkedin feed, we can leave the tokens there

even for a year, that we can rotate the tokens once in every six months or once in every year.

11) How to use Tokens with Frontends

While working with JavaScript front ends that were using restful APIs through

Oauth Where did you store the Oauth token on the frontend?

We stored them on the server that is hosting the Front end and in some cases, if we had to send this token to the Web browser, for some reason we have used tls that is the transport layer security to ensure that there is no unwanted access.

–

12) What is CSRF

CSRF stands for cross site request forgery.

When we are browsing through a website, that website might set some cookies onto the Web browser to maintain that user session.

At the same time, if the user is also browsing through another website, this website can steal the cookies that were sent by the other website and use those cookies to forge excession.

It can send in a request by forging a session and it can steal our data. The user data that is there on this website on the server that process is called cross site request forgery.

---

13) How to prevent CSRF

How did you prevent cross site request forgery while using spring security for your Web applications?

When we use spring security for our Web applications automatically, spring security will prevent CSRF from happening.

There is a CSRF filter that will generate a token and it will include the token in every Web page that goes back to the client.

So when the client tries to submit a form or a request, that CSRF token should come back to the server.If it doesn't happen, the CSRF filter will reject that request.

So even if another hacking website, a forging website tries to steal the browser session and it tries to submit a form on behalf, just like acting as if it is the original website, that guy will not have the generated CSRF token, only the Web application, real web pages will have the CSRF token.
So that request will be rejected.That is how this CSRF filter in spring security

works. 14) What is CORS and how did you use it in your application?

CORS stands for Cross Origin Request Sharing that is when we have our back ends running on one domain, on one server with its own IP address and domain name and front ends on a different domain, port number, etc. will have issues when this Front end, whether it is building Angular, react, etc., tries to communicate with this restful API on the back end,which is running on a different domain. That is where cors comes in.

If you do not configure cors for application, you will see cross origin request errors once you configure cors using the simple annotations in spring like @ cross origin.

These applications will be able to communicate with each other seamlessly and the front end will be able to make restful calls and get the responses back.

Cors allows us to configure the security at origin level, the HTTP methods that are allowed and the HTTP headers that are allowed, all that can be configured while using cors.

Using this annotation, you can specify which application, front end running on which domain can communicate with the back end.

That is the origin http methods like get put posts which are allowed from this cors origin or cors domain application that is also possible.

And then at the http headers also, you can specify what headers can be included when that cors origin request comes in.

Java Messaging Service

1) What is messaging

Messaging is the process of exchanging business data across applications or across components within a application.

We have a sender of the message, we have the receiver of the message, and we have a messaging server or Broker, like active MQ, Sonic, MQ WebSphere, MQ, Tibco MQ and many more.

It is the responsibility of the messaging server to receive the message and ensure that it is delivered to a receiver.
-

2) Why Messaging

Messaging offers various advantages, starting with heterogeneity, that is, these applications that use messaging can be written in different programming languages and can run on different operating systems, and they can communicate with each other through their messaging brokers and the virtual channels, like queues and topics they are loosely coupled because they don't have to know about each other.

They communicate through the messaging server and they are completely loosely coupled, which offers us flexibility and agility to replace one micro service application that is using messaging with another micro service application without affecting any other micro service applications using it since they are loosely coupled, last and very important to reduce bottlenecks through asynchronous communication.

That is, the sending micro service or an application need not wait for the response to come back.

Once it puts a message, it can continue doing its work and whenever there is a response, it can take over and use that response.

Also, as the message load increases, we can easily scale by increasing the number of receivers or consumers.

All these advantages make messaging a key part of any architecture, especially a micro services architecture.

-

3) What is JMS

JMS stands for Java Messaging Service.

It is a Java EE standard to use messaging in Java applications just like any other Java EE standard. It comes with a specification and API.

The specification is a set of rules written in plain English, and it is for the messaging brokers or messaging servers like Active MQ, Web Logic, WebSphere, etc..The API is for the developers.

Once we learn the JMS API, we don't have to learn each of these messaging brokers apis before JMS, If we want to use any of these message brokers and communicate to them from our applications, we had to learn these brokers like ACTIVE MQ WebSphere, their own apis.
Now we have one standard we Master the JMS API, a few interfaces, classes and their methods and will be able to work with any of these messaging servers or brokers from within our application to do messaging.

4) What is the KEY JMS API

What are the key interfaces or classes in the JMS 2.0 API and what are the steps to send and receive messages?

The key classes and interfaces are the connection factory the JMS context and then using the JMS context, we can create a consumer or a producer and start producing and consuming messages.

And the typical steps are to look up for a connection factory, the messaging broker or the messaging service like WebSphere,Active MQ, etc. They will create the connection factory and they will put it in the JNTI.

Once we have access to the JNTI, we can get that connection factory we create a session or A JMS context depending on the version of JMS we are using.Look up for a destination which is a queue or a topic, and then send and receive messages.

So we will get the initial context, which is the point to the Jnti, we then look up for the Queue or a topic.We also get the JMS context by getting the connection factory.

Once we have the JMS context, you can create producers, consumers and start sending and receiving messages.

---

5) Two Types of messaging

Two types of messaging are point to point and publish subscribe. Point-to-point, also known as P2P, is where the messages are exchanged using a virtual channel called Queue. The center of the message is called producer.

The receiver is called a receiver or a consumer.

Once a consumer consumes the message from the Queue, it is gone from the Queue and no other consumer will be able to read it.

P2P supports asynchronous fire and forget as well as Synchronoss request and reply messaging as well. P2P is like sending email.
Once the email is received, it's gone. No, it will be received only by one person or one consumer publish and subscribe is where the messages are exchanging that using a virtual channel called Topic.

And here the sender of the message is called the producer. And there are multiple consumers known as subscribers.

The messages will be broadcasted to all the subscribers, like a newspaper subscription. -

6) JMS Transactions

How can we use transactions in JMS producers and consumers?

when we create a JMS context object, we can enable transactions using JMS context, dot session, underscore transacted.

If we pass this parameter, then a transaction will be started as we produce the messages.

All those messages will not be committed to the queue until we issue a JMS context dot commit and they will not be visible for the consumers to consume them.

Only when you issue a JMS context dot commit then the consumers will be able to consume those messages and they will be visible to them.

If we want to roll back, we can use JMS as context dot rolled back.

Similarly, on the consumer side as well, when we create the JMS context, we enable the transaction and we can use JMS commit or JMS roll back to read all the messages.

Only once we do a JMS commit all the messages will be gone from the queue.

If we do roll back, those messages will still be available on the queue for other consumers or for the same consumer to consume them later on.

---

7) What is Message Grouping?

As the messages are received on a queue, they will be consumed by different consumers, but if we have a scenario where we want multiple messages that are being received to be consumed by the same consumer, we use the concept of message grouping will group all these messages by setting a header on these messages and which messages have the same header value will be consumed by that same consumer, that JMS provider is to ensure that they are consumed by the same consumer.

So when you create the messages, you will set a string property on that message. Call JMSX group id and provide a value which all messages will have the same JMSX group id value will be consumed by the same consumer.

-

8) What is a MDB

MDB stands for Message Driven Bean, it is a part of the EJB standard and any application servers like Web logic WebSphere JBoss, which has support for EJB, has support for MDBs, MDBs are java classes

They are just like message listeners in JMS. We create a MDB by implementing a interface and configure it using xml or annotations.

Once we do that, whenever there is a message on a particular Queue, automatically these applications servers like JBoss will trigger the MDB and they will hand over that message to that MDB and it can handle those messages asynchronously.
–

Design Patterns

1) What are the Design Patterns you have used in your projects?

At a minimum, we would have used the following patterns, the data access object pattern to implement our data access layer.

It's a simple class, where we rate all the data access code, A Singleton pattern to implement utility classes where only one instance of a particular class should be created, the prototype pattern indirectly or directly.

When we use frameworks like spring, we have the option to configure singleton beans or prototype beans for our application, the dependency injection and IOC patterns when we work with spring and the dependency injection across the layers.

The MBC pattern for our web layer, especially if we are using spring web, factory pattern if you are using utility classes and using factories to create the instances of those classes.

In addition to this, you can specify as many design patterns as you know, and you have to use it like facade builder, command, etc. but be ready to answer any questions on these patterns you mentioned.

--