|  |
| --- |
| Question |
| What are main uses of Stream API ? Explain Differences between Collection API and Stream API? |
| What is a DeadLock and RACE condition ? How do you propose to fix it ? |
| What are the non functional requirements you would focus on while working on a task ? |
| Explain the term ‘Statelessness’ with respect to RESTful WEB service. |
| What are the best practices to follow for exception handling ? (How do we work with Global Exception Handling ? |
| How is performance testing done in your project ? Can you explain the testing cycle in the project. |
| What testing strategy have you used in your project ? What is Testing Pyramid and different stages . |
| What do you understand by the Spring IoC Container? Explain their types. |
| What would you do if your changes fails in production and you have dependant applications failing as well ?  What would your mitigation plan be ? |
| How is DevOps different from agile methodology? What are the different phases in DevOps ? |
| What is the most challenging task that you have worked on ? |
|  |
|  |
|  |
| quark |
|  |
| What would you do if you notice one of your new team member refuses to write unit test? |
| What would you tell him / her, how does a good unit test look like? (FIRST)  What is the difference between fast and timely? Mock vs spy? |
| What is the difference between CI/CD? How does the CI/CD pipeline work on your project? What would you improve? |
| What is java garbage collector? Why is it important? |
| What design patters does Spring framework use and provide to  developer? |
| What is dependency injection? Do you use it during your everyday work? What are the benefits of it? |
| How would you investigate a production issue? For example, when the application is crashed with no logs |
| How would you conduct code review? Have you met with static code analysis tools? What metrics do you use in your project? |
| What does transaction mean in for you? What ACID stands for?  Could you describe  Durable and Isolated properties? |
| NoSQl vs Relational databases? How would you choose database for your application?  Which principles do you use for it? |
| What is the difference between functional and non-functional requirements?  What are the practices do exist in requirements tracking? |
|  |
| +1 What would you do if you notice the burnout on one of your employee? |
|  |
| removed qusetions |
| What are the practices to follow when you design RESTful API? (What is Richardson Maturity Model (RMM)? ) |
| What are spring beans? What are their scopes?   |  |  |  | | --- | --- | --- | | Tamas Ozsvath | soft skill | What would you do if you notice one of your new team member refuses to write unit test? | | Tamas Ozsvath | testing | What would you tell him / her, how does a good unit test look like? (FIRST) What is the difference  between fast and timely? Mock vs spy? | | Tamas Ozsvath | devops / practical | What is the difference between CI/CD? How does the CI/CD pipeline work on your project?  What would you improve? | | Tamas Ozsvath | java | What is java garbage collector? Why is it important? | | Tamas Ozsvath | java | What design patters does Spring framework use and provide to  developer? | | Tamas Ozsvath | java / spring | What is dependency injection? Do you use it during your everyday work? What are the benefits of it? | |  | java / practical | How would you investigate a production issue? For example,  when the application is crashed with no logs | | Tamas Ozsvath | practical | How would you conduct code review? Have you met with static code analysis tools?  What metrics do you use in your project? | | Tamas Ozsvath | db | What does transaction mean in for you? What ACID stands for?  Could you describe Durable and Isolated properties? | | Tamas Ozsvath | db | NoSQl vs Relational databases? How would you choose database for your application?  Which principles do you use for it? | | Tamas Ozsvath | testing | What is the difference between functional and non-functional requirements?  What are the practices do exist in requirements tracking? | |  |  |  | | Tamas Ozsvath | soft skill | +1 What would you do if you notice the burnout on one of your employee? | |

What is the most challenging project / task ?

What are the possibilities and limitations of a tool / technology / approach ?

What is the difference between java 1.5 and java 1.8 .

What is the best way to do something ?

Messaging

"1. Share your experience with messaging system(in CV it's told he had experience with IBM MQ)

2. What type of communucation messaging is provided: sync or async?

3. Compare sync vs async communication

4. What is the JMS?

5. What types of messaging models are provided by JMS?

6. Durable vs non-durable subscriber?

7 What is the idempotent consumer?

8 What Delivery guarantees are provided by IBM MQ"

software development methodologies

"1. What software methodologies do you know?

2. What are the advantages of Agile over Waterfall?

3. SCRUM vs KANBAN, differencies, how to choose?

4. What estimation approach did you choose/know for story/task estimations?

5. What SCRUM ceremonies do you know?"

CI/CD

"1. Describe CI/CD process on your last project

2. Describe in details how deployment process is organized?

3. What deployment strategies do you know?

4. TDD and BDD

5. Acceptance testing vs end-to-end testing"

Microservices

"1. What is that main pros and cons of a Microservices Architecture in comparison to other architectural styles?

2. What is the API Gateway?

3. What is the eventual consistency?

4. Database per service or one database for all services?

5. CQRS"

1 Java What did you use to handle continuous and unbounded data set in Kafka ?

2 Java What is a DeadLock and RACE condition ? How do you propose to fix it ?

3 Java What are the non functional requirements you would focus on while working on a task ?

4 Java Explain the term ‘Statelessness’ with respect to RESTful WEB service.

5 Java What are the best practices to follow for exception handling ? (How do we work with Global Exception Handling ?

6 Java Why are immutable classes generally the best candidates for Hashmap keys?

Java What design patters does Spring framework use and provide to  developer?

7 Testing "What testing strategy have you used in your project ? What is Testing Pyramid and different stages.

What is the basic rule for writing a good unit test look like? (FIRST)

What is the difference between fast and timely? Mock vs spy?"

8 Prac "What would you do if your changes fails in production and you have dependant applications failing as well ?

What would your mitigation plan be ? For example, when the application is crashed with no logs. "

9 Database "What does transaction mean in for you? What ACID stands for? Could you describe

Durable and Isolated properties?"

10 practical "How would you conduct code review?

Have you met with static code analysis tools?

What metrics do you use in your project?"

11 Java "What is thread pool executor ?

Why is it advisable to use a thread pool executor custom implementation, rather than a fixed or cached thread pool executor from a built-in service?"

12 Messaging "1. Compare sync vs async communication? Provide pros and cons

2. Describe for what purposes did you use Kafka on your projects?

3. Compare Kafka and traditional queues, what is the main difference beetween them

4. In what usecase what you would advise to use?

5. What is idempotent consumer pattern? What is the idempotent producer in Kafka?

6. What delivery semantics are provided by kafka?"

13 Cloud(AWS) "1. What are the advantages of Cloud againts traditional on-premise?

2. Types of Cloud services(IAAS, FAAS,PAAS), difference between them, examples in AWS

3. Serverless, pros&cons. What are the restrictions?

4. Could you describe briefly what EC2 service in AWS?

5. Do you know the difference beetween difference between virtualization and containerization?

What are the benefits of containers agains VMs

6. SQS vs SNS"

14 Databases "1. Provide the key differences beetween SQL and NoSQL databases

2. BASE

3. Replication and sharding

4. Types of NoSQL(document, columnar, graph, key-value)"

15 software development methodologies "1. What software methodologies do you know?

2. What are the advantages of Agile over Waterfall?

3. SCRUM vs KANBAN, differencies, how to choose?

4. What estimation approach did you choose/know for story/task estimations?

5. What SCRUM ceremonies do you know?

6. TDD and BDD"

|  |
| --- |
| What is thread pool executor ? Why is it advisable to use a thread pool executor custom implementation, rather than a fixed or cached thread pool executor from a built-in service?  You have multithreaded app which looks properly synchronized, but sometimes it seems that when particular field is read it returns old value. You suspect some synchronization issue, how would you investigate? |
|  |
| What do you mean by Cohesion and Coupling in a Microservices driven application ?c What does one mean by Load Balancing ? How is it implemented in Spring Cloud.  How to achieve server side load balancing using Spring Cloud |  |
| What is an IoC container in Spring ? Why does Application context use Aggressive Initialization ?  What are the different ways to create Spring Beans thread safe ? |  |
| Transaction processing, like I buy a product from eCommerce which will involve multiple microservice to call one by one say Order, Payment and Notification micro service involved how to revert a transaction if it fails at any level |  |
| What are the non functional requirements you would focus on while working on a task ? |  |
| Explain continuous integration model being used in your project. |  |
| How do you do a release in your project. Explain your release strategy and brancing strategy. What is the difference between CI/CD? How does the CI/CD pipeline work on your project? What would you improve? |  |
| What testing strategy have you used in your project ? What is Testing Pyramid and different stages. |  |
| What is the basic rule for writing a good unit test look like? (FIRST) |  |
| What is the difference between fast and timely? Mock vs spy? |  |
| What would you do if your changes fails in production and you have dependant applications failing as well ? What would your mitigation plan be ? For example, when the application is crashed with no logs. |  |
|  |  |
| What does transaction mean in for you? What ACID stands for? Could you describe |  |
| Durable and Isolated properties? |  |
| Explain the term ‘Statelessness’ with respect to RESTful WEB service.  Can we send an Custom Object as the response of Controller handler method What is the upper limit for a payload to pass in the POST method |  |
| What are the best practices to follow for exception handling ? (How do we work with Global Exception Handling ? |  |
| How would you conduct code review? |  |
| Have you met with static code analysis tools? |  |
| What metrics do you use in your project? |  |

|  |
| --- |
| Could you please explain when Kanban is better suited for a team than Scrum/Agile? |
| What is Java Virtual Machine? What components of JVM do you know? Why Java is called "Platform independent programming language"? What is Java Memory Model? |
| Could you please describe the purpose and use-сases of the Fork/Join Framework in Java? |
| Please explain Java Collection Framework hierarchy. What is the difference between HashMap and TreeMap? How does TreeMap work? What is Big O notation? |
| What is atomic operation? What atomic classes are you familiar with in Java Concurrency API? |
| What architecture styles are you familiar with? What is event-driven architecture?  How does event-driven architecture differ from other architectural styles like microservices or SOA? |
| Could you please explain the architecture of a system you have built on LBTM-WCDC (compensation claims management system) project? |
| What is API gateway? What main parts does it consist of? How the load balancing is done when API gateway is used? |
| Let's say you have a backend service with 100 instances deployed across multiple data centers and you have a load balancer (nginx or anything else) in front of them. How would you load balance a stateful HTTP request in this case? |
| What does it mean for you to be a Senior Software Engineer? |
|  |
|  |
|  |
| What is the role of an IoC container in Spring ? Why does Application context use Aggressive Initialization ?  What are the different ways to create Spring Beans thread safe ? |
| Why are streams lazy in nature ? Can you explain how parallel streams work in Java 8 ? |
| Explain the term ‘Statelessness’ with respect to RESTful WEB service.  Can we send an Custom Object as the response of Controller handler method What is the upper limit for a payload to pass in the POST method |
| In your work for Invoice Creation Servics , how did you optimize the calculation logic and criterias for the bill generation.  How to handle the security for the same. |
| Can you explain where have you used Aurora ?  Is it possible for the application to fail over to the cross-region replica from the current primary .  What are the options for scaling the compute resources associated with my Amazon Aurora DB Instance |
| What would you tell him / her, how does a good unit test look like? (FIRST)  What is the difference between fast and timely? Mock vs spy? |
| What is the difference between CI/CD?  How does the CI/CD pipeline work on your project?  What would you improve? |
| What does transaction mean in for you? What ACID stands for?  Could you describe Durable and Isolated properties? |
| What is the most challenging task that you have worked on ? |
| What would you do if your changes fails in production and you have dependent applications failing as well ? What would your mitigation plan be ? For example, when the application is crashed with no logs. |

|  |
| --- |
| Let’s imagine a situation that you have a microservices application and at some point a customer complains that 1 of the features works very slowly, and you should improve it. How would you start working on a task like that? Let's say a problem was found in microservices communication/sql/heap overflow |
| How does Spring transaction management work under the hood? (TransactionInterceptor Aspect). Spring AOP, proxy types |
| What steps do you need to take to ensure the best possible code quality in the project? |
| Let’s image your current project uses waterfall methodology and a customer asks you to set up an agile methodology instead. Which one would you choose and why? What steps would you take in order to set it up? |
| You highlighted in your CV that you have some experience in leading some of the development activities. Could you please elaborate on this. What was your role in these activities? |

|  |
| --- |
| What is the role of an IoC container in Spring ? Why does Application context use Aggressive Initialization ?  What are the different ways to create Spring Beans thread safe ? |
| Why are streams lazy in nature ? Can you explain how parallel streams work in Java 8 ? |
| As you mentioned you have worked on improving the availability on the application can you explain how did speed up Spring Boot Startup Time. Any JVM tweeks perhaps . |
| How to enhance a Spring Boot REST API to deal with tens of thousands or more incoming requests? Can we send an Custom Object as the response of Controller handler method What is the upper limit for a payload to pass in the POST method |
| What would you do if your changes fails in production and you have dependent applications failing as well ? What would your mitigation plan be ? For example, when the application is crashed with no logs. |
| How does the CI/CD pipeline work on your project?  What would you improve? |
| How does a good unit test look like? What is the difference between fast and timely? Mock vs spy? |
| What do you mean by Cohesion and Coupling in a Microservices driven application ? |
| You have experience with multiple frameworks . Could you please explain when Kanban is better suited for a team than Scrum/Agile? |
| How to achieve server side load balancing using Spring Cloud ?  Let's say you have a backend service with 100 instances deployed across multiple data centers and you have a load balancer (nginx or anything else) in front of them. How would you load balance a stateful HTTP request in this case? |

Node JS Questions :

548 577 5119

[12:32 a.m.] Saikiran Kondru

---> Why Node.js is single threaded?

Node.js uses a single threaded model in order to support async processing. With async processing, an application can perform better and

is more scalable under web loads. Thus, Node.js makes use of a single-threaded model approach rather than typical thread-based implementation.

---> How does concurrency work in Node.js?

The thing with node.js is that everything runs concurrently, except for your code.

So, what that means is that there are actually lots of threads running inside Node.js virtual machine (or a thread pool if you wish),

and those threads are utilized whenever you call an async function like performing i/o operations on files, accessing databases,

requesting urls, etc.

However, for your code, there is only a single thread, and it processes events from an event queue. So, when you register a callback its reference

is actually passed to the background worker thread, and once the async operation is done, new event is added to the event-queue with that callback

When Node gets I/O request it creates or uses a thread to perform that I/O operation and once the operation is done, it pushes the result to

the event queue. On each such event, event loop runs and checks the queue and if the execution stack of Node is empty then it adds the queue

result to execution stack.

[12:32 a.m.] Saikiran Kondru

--> What is package.json?

The package.json file in Node.js is the heart of the entire application. It is basically the manifest file that contains the metadata

of the project where we define the properties of a package.

[12:33 a.m.] Saikiran Kondru

---> Explain the difference between local and global npm packages installation

The main difference between local and global packages is this:

local packages are installed in the directory where you run npm install <package-name>, and they are put in the node\_modules folder under

this directory

global packages are all put in a single place in your system (exactly where depends on your setup), regardless of where you run

npm install -g <package-name>

In general, all packages should be installed locally.

This makes sure you can have dozens of applications in your computer, all running a different version of each package if needed.

Updating a global package would make all your projects use the new release, and as you can imagine this might cause nightmares in terms

of maintenance, as some packages might break compatibility with further dependencies, and so on.

[12:33 a.m.] Saikiran Kondru

How the Event Loop Works in Node.js?

The event loop allows Node.js to perform non-bocking I/O operations despite the fact that JavaScript is single-threaded.

It is done by offloading operations to the system kernel whenever possible.

Node.js is a single-threaded application, but it can support concurrency via the concept of event and callbacks.

Every API of Node.js is asynchronous and being single-threaded, they use async function calls to maintain concurrency.

Node uses observer pattern. Node thread keeps an event loop and whenever a task gets completed, it fires the corresponding event which signals the event-listener function to execute.

Features of Event Loop:

Event loop is an endless loop, which waits for tasks, executes them and then sleeps until it receives more tasks.

The event loop executes tasks from the event queue only when the call stack is empty i.e. there is no ongoing task.

The event loop allows us to use callbacks and promises.

The event loop executes the tasks starting from the oldest first.

[12:36 a.m.] Saikiran Kondru

What are Modules in Node JS?

[12:36 a.m.] Saikiran Kondru

Modules are JavaScript libraries you can include in your project.

Consider modules to be the same as JavaScript libraries.

A set of functions you want to include in your application.

\*\*\*\*\*\*\*\*\*\* Built in Module that you worked -- Events, streams

--> ---> Node.js as a File Server

The Node.js file system module allows you to work with the file system on your computer.

To include the File System module, use the require() method:

var fs = require('fs');

[12:37 a.m.] Saikiran Kondru

What do you understand by an Event Emitter in Node.js?

EventEmitter is a Node.js class that includes all the objects that are capable of emitting events.

These objects contain an eventEmitter.on() function through which more than one function can be attached to the named events that

are emitted by the object. Whenever an EventEmitter object throws an event, all the attached functions to that specific event

are invoked synchronously. Below code shows how to us the EventEmitter in your application:

[12:37 a.m.] Saikiran Kondru

What is an error-first callback in Node.js?

Error-first callbacks in Node.js are used to pass errors and data.

The very first parameter you need to pass to these functions has to be an error object while the other parameters represent the associated data.

Thus you can pass the error object for checking if anything is wrong and handle it. In case there is no issue, you can just go ahead and with

the subsequent arguments.

[12:37 a.m.] Saikiran Kondru

What do you understand by callback hell? How would you fix that?

Callback Hell is also known as the Pyramid of Doom. It is a pattern caused by intensively nested callbacks which are unreadable and unwieldy.

It typically contains multiple nested callback functions which in turn make the code hard to read and debug.

It is caused by improper implementation of the asynchronous logic.

How can you avoid callbacks?

To avoid callbacks, you can use any one of the following options:

You can use modularization. It breaks callbacks into independent functions.

You can use promises.

You can use yield with Generators and Promises.

[12:37 a.m.] Saikiran Kondru

Why would you prefer promises over callbacks?

You must answer such Node.js interview questions with a relevant yet precise reason explaining why you would choose one over the other.

The reasons for selecting promises over callbacks are as follows:

Promises easily handle multiple asynchronous operations providing better error handling than callbacks and helping in avoiding the

undesired callback hell situation.

Built-in error handling

The coupling is low.

Improved readability

[12:38 a.m.] Saikiran Kondru

Explain what is Reactor Pattern in Node.js?

What do you understand by Reactor Pattern in Node.js?

Reactor Pattern in Node.js is basically a concept of non-blocking I/O operations. This pattern provides a handler that is associated with

each I/O operation and as soon as an I/O request is generated, it is then submitted to a demultiplexer.

This demultiplexer is a notification interface which is capable of handling concurrency in non-blocking I/O mode.

It also helps in collecting each and every request in the form of an event and then place each event in a queue.

Thus resulting in the generation of the Event Queue. Simultaneously, we have our event loop which iterates the events present in the Event Queue.

---> Does Node.js provide any Debugger?

Node.js do provide a simple TCP based protocol and debugging client that comes built-in. In order to debug your JavaScript file,

you can use the below debug argument followed by js file name that you want to debug.

[12:38 a.m.] Saikiran Kondru

What do you understand about node.js streams?

Node.js streams are instances of EventEmitter. You can use them to work with streaming data in Node.js,

especially for handling and manipulating streaming large files (such as videos and mp3) over the network. Streams use buffers for temporary storage.

There are four main types of streams.

Writable streams to write data (e.g. fs.createWriteStream())

Readable streams to read data (e.g. fs.createReadStream())

Duplex streams that are readable and writable (e.g. net.Socket)

Transform streams or duplex streams can modify or transform the data as it is written and read (e.g., zlib.createDeflate())

[12:38 a.m.] Saikiran Kondru

How would you enhance Node.js performance through clustering?

Node.js applications do not use multiple core systems. Instead, they run on a single processor.

In Node.js, the cluster mode helps startup multiple node.js processes, thus having multiple instances of the event loop.

When you use a cluster in a Node.js app behind the scenes, there are numerous node.js processes created,

but there is also a parent process called the cluster manager. A cluster manager monitors the health of the individual instances of the application.

[12:39 a.m.] Saikiran Kondru

What is the difference between Asynchronous and Non-blocking?

1. Asynchronous:

The architecture of asynchronous explains that the message sent will not give the reply on immediate basis just like we send the mail but

do not get the reply on an immediate basis. It does not have any dependency or order. Hence improving the system efficiency and performance.

The server stores the information and when the action is done it will be notified.

2. Non-Blocking:

Nonblocking immediately responses with whatever data available. Moreover, it does not block any execution and keeps on running as per the requests.

If an answer could not be retrieved then in those cases API returns immediately with an error. Nonblocking is mostly used with I/O(input/output).

Node.js is itself based on nonblocking I/O model. There are few ways of communication that a nonblocking I/O has completed.

The callback function is to be called when the operation is completed. Nonblocking call uses the help of javascript which provides a callback

function.

[12:39 a.m.] Saikiran Kondru

What is difference between promises and async-await in Node.js?

1. Promises:

A promise is used to handle the asynchronous result of an operation. JavaScript is designed to not wait for an asynchronous block of code to completely execute before other synchronous parts of the code can run. With Promises, we can defer the execution of a code block until an async request is completed. This way, other operations can keep running without interruption.

States of Promises:

Pending: Initial State, before the Promise succeeds or fails.

Resolved: Completed Promise

Rejected: Failed Promise, throw an error

Example:

function logFetch(url) {

  return fetch(url)

    .then(response => {

      console.log(response);

    })

    .catch(err => {

      console.error('fetch failed', err);

    });

}

2. Async-Await:

Await is basically syntactic sugar for Promises. It makes asynchronous code look more like synchronous/procedural code, which is easier for humans to understand.

Putting the keyword async before a function tells the function to return a Promise. If the code returns something that is not a Promise, then JavaScript automatically wraps it into a resolved promise with that value. The await keyword simply makes JavaScript wait until that Promise settles and then returns its result.

Example:

async function logFetch(url) {

  try {

    const response = await fetch(url);

    console.log(response);

  }

  catch (err) {

    console.log('fetch failed', err);

  }

}

How to achieve server side load balancing using Spring Cloud ?

Route 53 can be used to route users to infrastructure outside of AWS ?

Can I use the same policy to manage routing for more than one DNS name?

How am I billed for using Amazon Route 53 Traffic Flow?

How Amazon Route 53 checks the health of your resources

Can you explain what the control flow of a typical user authentication process looks like in AWS Cognito?

What’s the best way to manage permissions when accessing an object stored on S3 while using Cognito?

What is the maximum number of login attempts allowed per second by the default configuration?

Let's say you have a backend service with 100 instances deployed across multiple data centers and you have a load balancer (nginx or anything else) in front of them. How would you load balance a stateful HTTP request in this case?

What is API gateway? What main parts does it consist of?

How the load balancing is done when API gateway is used?

1. What are the advantages of Cloud againts traditional on-premise?

2. Types of Cloud services(IAAS, FAAS,PAAS), difference between them, examples in AWS

3. Serverless, pros&cons. What are the restrictions?

4. Could you describe briefly what EC2 service in AWS?

5. Do you know the difference beetween difference between virtualization and containerization?

6. What are the benefits of containers agains VMs

Let’s imagine a situation that you have a microservices application and at some point a customer complains that 1 of the features works very slowly, and you should improve it. How would you start working on a task like that? Let's say a problem was found in microservices communication/sql/heap overflow