Below are the requirements we are looking for:

* Proficiency in Java: Demonstrated expertise in Java programming, with a focus on automation-related tasks and projects.
* In-depth Understanding of OOP Principles: Strong grasp of object-oriented programming concepts, enabling the development of maintainable and efficient automation code.
* Familiarity with CI/CD Tools: Experience with Jenkins or similar CI/CD tools, including setting up and managing automated testing and deployment pipelines.
* Strong Problem-Solving Skills: Proven ability to identify, analyze, and resolve issues in automation scripts efficiently and effectively.
* Experience with Version Control (Git): Proficiency in Git for version control, including branching, merging, and resolving conflicts.
* JUnit and TestNG Tests: Experience in designing and executing JUnit tests for automated unit testing and TestNG tests
* Selenium: Hands-on experience with Selenium for web automation testing.
* Gradle and Maven: Familiarity with build automation tools like Gradle and Maven for project management and dependency resolution.

Fifty percent of the team:

* At least 4+ years of technical project consulting and implementation experience in SharePoint and Nintex.
* Moderate to deep expertise with relevant Microsoft technologies in the marketplace, e.g., SharePoint, M365, Power Platform, PowerShell, Dynamics, Azure, and related development technologies.
* Deep expertise with relevant Nintex technologies, e.g., Nintex Workflow, Nintex Forms, etc.
* Experience with developing and implementing solutions using PowerShell, JavaScript, jQuery, etc.
* Azure Cloud deep understanding from the developer in DevOps point of view.

Fifty percent of the team:

* At least 4+ years of experience with C# and .NET
* Familiar with Visual Studio or Rider and Git version control
* Expertise in the development of REST APIs and web applications (ASP.NET Core, Blazor)
* Exposure to a cloud computing platform (Microsoft Azure, AWS)
* Knowledge of CI/CD systems (Azure DevOps, Jenkins, TeamCity)
* competence in Angular will be a bid plus

1. Senior Android Developer with proven track record in building scalable mobile applications

* Proficient knowledge of Kotlin, Coroutines, Koin, Retrofit, and MVVM with clean architecture.
* Record in product and design methodologies and data-informed development
* Experience in building Streaming experiences
* Commercial experience with AWS

2. Senior React Native Dev with at least 5 years of experience in consumer application development with a focus on React Native.

* Expert in JavaScript, TypeScript, and related frontend technologies such as Redux, MobX.
* Developing applications that use RESTful APIs.
* Completed projects with native modules and integrating with native libraries.
* Knowledge of video codecs and streaming protocols.

Example Questions during the interview

Go over your CV - self presentation.  
Question about the architecture of one of the latest projects.

-- following questions are often based on something mentioned by you - in this case Spring and being BE dev in the last project  
-- from time to time they switch the direction

* Spring Boot - what is the 'boot' part? How do we use it?
* Dependency container and injections - can you tell us something about them?
* What is the reason for using dependency injections while writing code? Is it useful in unit testing?
* What is benefit of unit test?
* Are you familiar with TDD?  
  How would you find dependency in maven/Gradle?
* How Direct dependencies and sub dependencies
* What is the approach for using Spring Security? How would you configure it? - they are looking for "security filter chain" answer.
* Which Java version have you used?
* Tell us about the basics of OOP.
* Tell us about inheritance in Java. Why are we using it?
* Tell us about encapsulation in Java. Access modifiers.
* Tell something about Garbage collector. How does it know that some objects are not used?
* What happens to the JVM and your programs while the GC is working? (waits)
* Difference between abstract class and interfaces?
* Functions in interface?
* Have you heard about functional interfaces? What is the definition?
* Have you used concurrent programming in Java? Related block?

DB  
What kind of joins do you know?  
What is the difference between left and right join?  
How would you improve the performance of your query? (indexes - why? how? when they are bad? stored procedures as they are precompiled)  
What is write ahead lock?

Tell me about RESTFul APIs that you have used.  
Which are the HTTP methods?

I saw you have Angular experience - which was the version?  
What is a component in Angular?   
Can you preload data in Angular before the component is shown?

At the end few minutes for your questions. Make sure you ask questions!

1) what is SOLID  
2) SQL vs NoSQL DB (what is the difference)  
3) different spring bean scope (<https://www.baeldung.com/spring-bean-scopes)>  
4) what is WebSocket  
5) component scanning in Spring  
6) design patterns - singleton, factory, builder, etc.

защо Java?

SAOP experience ? difference between REST and SOAP ? - xml, wsdl

SOLID

last java version used? - new features?

difference between map and flat map

streams - which are the most noticeable limitations ? - final?  
what is functional interface ?

design patterns   
- what is factory pattern?  
- what is adapter pattern?

IoC

multi-threading  
- which are the different thread states?

HashMap - how does it work internally?

Why would we want to overwrite the hash method?

difference between controller and rest controller in spring?

>Роля в проектите по които съм работил  
>OOP-Polymorphism  
>SOLID  
>CLR (common language runtime)  
>CLS (common language specification)  
>GC (garbage collector)  
>Class vs Struct  
    + Reference vs Value type  
    + ref/out keywords  
>Records  
>Design patterns който съм използвал  
>Видове lifecycle на сървиси в DI в .NET core  
>Начини по които е възможно да се ресолвне регистриран сървис  
>Delegate и вече дефнирани видове делегати в C# (Action, Func)  
>опит с Azure  
>опит с Entity Framework  
>(понеже си работил с MS SQL и PostgreSQL) Каква е разликата между двете и коя е по-добра?  
>First() vs Single() в linq  
>LEFT JOIN в SQL  
>UNION vs UNION ALL в SQL  
>За какво се използват Transactions в SQL  
>Stored procedures vs Functions в SQL  
>Indexes в SQL  
>Писал ли съм unit тестове?  
>Работил ли съм по някоя Agile методология?

питат за setup-a на екипа с който си работил, за процеса на работа (дев, тест, среди)

4 принципа на ООП - абстракция, енкапсулация, полиморфизъм, inheritance  
access modifiers в java (private, protected, ...)  
overload vs override  
interface vs abstract class - koga koe

структури от данни - list, arrayList, linkedList, Set, HashSet, HashMap  
какво си ползвал от спринг  
DI  
SQL  
как правиш many-to-many v DB-to  
REST  
post vs put  
put vs PATCH  
token? jwt ?  
git  
merge vs rebase  
работил ли си със cloud?  
design patterns  
какво искаш да правиш? какво ти е интересно? как искаш да се развиваш?

what is bad or good architecture  
- different/separate components  
- documentation  
- unit tests

What is SOLID

what is reflection and when it is used

OOP design patterns  
- singleton  
- factory  
- builder  
- prototype  
<https://www.tutorialspoint.com/design_pattern/index.htm>

different spring bean scope (<https://www.baeldung.com/spring-bean-scopes)>  
component scannign in Spring

array vs list  
- dynamic vs static

pass by value vs pass by ref - in js only primitives are passed by value

scope of variables in js, hoisting   
closures

what is websocket

sql vs nosql db (what is the difference)  
sql: joins (left,right,full,inner)

git vs cvs/svn (distributed)

OOP - как ги разбираш  
access modifiers  
method signature - is valid same params different return type  
(boring theoretical questions)

structures of data - what is List, different implementations  
ArrayList vs LinkedList -> какво става като махмен един елемент  
Set - как се сравняват обекти? hashcode + equals - защо и двете?

Какво прави HashMap при колизия?  
Spring   
DI   
Hibernate, JPA, Spring Data Repositories  
Many-to-many in DB?  
Migration tools

Spring boot - is it microservices?  
комуникация между microservices?

REST contract - swagger  
REST + http methods

Design patterns  
  как правим Singleton (Spring или сами)? кога е необходим?  
  Factory?  
  Builder?   
  lombok?   
  
Какво искаш да правиш за напред?   
  Какви технологии?   
  Frameworks?   
  DB? Oracle PL/SQL?   
  Infrastructure?