Hypercontext

Interviews

Generated by Hypercontext.com

☐ DEVOPS ✓Expertise with various technologies (to be adjustable to alternative tools). ✓Mentoring software developers in gaining experience and assuming DevOps responsibilities. ✓Managing the installation and configuration of solutions. Not only doing maintenance and monitoring. For example, creating a Kubernetes cluster. ✓Building infrastructure from scratch, for example, migrating from a monolithic into a microservices architecture, from on-premise to Cloud, implementing Kubernetes. ✓Being open to new and better approaches. What are the red flags for the DevOps Engineer role at Proxify? ✖No experience with scripting or programming tools: "I did not have a chance to use it. Developers usually do scripting for me" ✖No experience with public Clouds: "I did not work with it because of security and sensitive data, but I know how Clouds work" ✖Limited experience with different Clouds: "I've been working with Clouds for 3 years: GCP (1 year), AWS (1.5 years), Azure (1 year). It is not a problem: all Clouds are the same" ✖No experience with Infrastructure as a Code tools: "I know the tools but rarely have to use them".

Summary:

Next Steps:

☐ NODEJS ❓OPEN QUESTION: Could you briefly describe your background, tools you've been working with, responsibilities, and types of projects? 🔎Industries: what field, sectors, or domains have you worked in? 🔎Languages: how long have you been working with JavaScript and Typescript in the field of web development commercially? 🔎Main skill: Why did you decide to start working with Node.js? How long have you worked with it commercially? 🔎Additional frameworks/libraries: What framework/library do you prefer to use with Node.js? How long have you had commercial experience with them? 🔎Clouds: which clouds do you have experience with? Which one do you prefer? how long have you worked with each cloud commercially? 🔎Infrastructure: have you worked with infrastructure? Building CI/CD pipelines or managing Docker or Kubernetes? 🔎Microservices: do you have experience with microservices architecture and migrations from monolith to microservice architecture? Which tools have you used? (note: smaller services, communication: express, Koa, DB to connect m-s, unit test, deployment, error handling, end-point) 🔎Databases: Which databases have you experience with? Have you used both SQL and NoSQL? Which ones specifically? When you go with SQL and when with NoSQL? 🔎Testing: what is your approach to testing? What tools do you use? 🔎Security: What are some security best practices you follow? (note: common security risks (e.g., XSS, CSRF, SQL Injection), practices like validating and sanitizing inputs, being aware of the OWASP top 10 vulnerabilities). What can assure the seniority of a Node.js developer? ✓Experienced in utilizing diverse frameworks and libraries based on project requirements. ✓Possesses a solid understanding of the entire development lifecycle. (best solutions) ✓Has hands-on experience in creating microservices architecture or executing migrations from monolithic to microservices. ✓Demonstrates a strong commitment to code quality and application maintainability. ✓Fundamental knowledge of Frontend and security best practices What are the red flags for the Node.js role at Proxify? ✖Only freelancing which is impossible to check the experience on real projects: "All of my projects have been as a freelancer, involving collaborations with diverse clients and multiple industries." ✖Not interested in testing the code: "I am not responsible for the code quality; we have a dedicated QA team that handles code testing and reports any bugs to us." ✖They can't explain clearly why they choose to work with the technology: "I chose to work with Node.js because it's currently one of the most in-demand and widely used technologies." ✖They don't

have experience using different libraries or frameworks: "I have a strong proficiency in JavaScript and Node.js. I am familiar with various libraries and frameworks, but I haven't found the need to use them"

Summary: Next Steps:

☐ VUE.JS ✓ Languages: proficiency in JavaScript and knowledge of TypeScript. ✓ Frameworks: experience with Vue3 and its new features (Composition API, setup function, Teleport); experience with SSR (Server Side Rendering Framework Like Nuxt.js) for an SEO requirement ✓ State Management: experience with state management solutions assures expertise in handling complex application states in large-scale projects. ✓ Performance Optimization: knowledge of techniques like code splitting, lazy loading, and minimizing re-renders reflects a senior developer's ability to create efficient and fast applications. ✓ Testing Expertise: proficiency in writing comprehensive unit tests and end-to-end tests for Vue.js components and applications using testing libraries like Vue Test Utils, Cypress, etc. ✓ Mentoring and Leadership: seniors often mentor and guide junior developers, provide technical leadership, and contribute to improving the development processes within a team. What are the red flags for the Vue.js Developer role at Proxify? ✖Lack of integration skills and knowledge of Vue.js Ecosystem: "I dont work much with libraries, but I have theoretical knowledge of Vuex". ✖Difficulty in explaining projects indicates a lack of depth in their experience: "I had no problems on my projects: it was mostly the maintenance phase." ✖Limited Knowledge of JavaScript (ES6+): "I mostly use older JS syntax and haven't fully explored the ES6+ features." ✖No Experience with API Integration: "I haven't had the chance to work with APIs in Vue.js projects. My projects mostly involve static data."

Summary: Next Steps:

☐ REACT JS Could you briefly describe your background, tools you've been working with, responsibilities, and types of projects? 🔎 Industries: what field, sectors, or domains have you worked in? 🔎 Framework: How long have you worked commercially with React.js? 🔎 Languages: How long have you worked with JavaScript and TypeScript? 🔎 Additional frameworks: Have you worked with any React.js frameworks or libraries beyond the core ones? (we can expect: Next.js, Gatsby.js, etc.) 🔎 State Management: How have you applied state management tools in React.js? Can you describe a case when you were using state management tools or had to choose between them? (we can expect: Redux, MobX, Recoil, Context API, Zustand, Apollo Client) 🔎 Integration with RESTful APIs: Can you walk me through your process of integrating APIs with a frontend application? What are key factors you take into account to make sure the integration goes smoothly? (we can expect: error handling, data normalization, caching strategies, optimist updates) 🔎 Styling: what CSS frameworks have you used for styling? (we can expect: Bootstrap, Material UI, Tailwind, etc.) 🔎 Testing: do you perform testing? what tools do you prefer for testing? (unit test - a must, E2E - plus) 🔎 Server-side rendering: Have you worked with server-side rendering? How and why? (we can expect: tools like Next.js or manual setups with ReactDOMServer, benefits like SEO improvements and better initial load performance.) 🔎 Why do you prefer to work with React instead of other frameworks like Angular, Vue.js? [optional]

Summary: Next Steps:

☐ ANGULAR 🔎 Industries: what field, sectors, or domains have you worked in? 🔎 Framework: How long have you worked commercially with Angular? 🔎 Languages: How long have you worked with TypeScript? 🔎 Additional frameworks: Have you worked with any Angular frameworks

beyond the core one? (we can expect ionic for mobile) 🔎State Management: can you share a case when you used state management tools or had to choose between them? (we can expect : NgRx, RxJS, Akita) 🔎Integration with RESTful APIs: Can you walk me through your process o f integrating APIs with a frontend application? What are key factors you take into account to m ake sure the integration goes smoothly? (we can expect: error handling, data normalization, ca ching strategies, optimist updates) 🔎Styling: what CSS frameworks have you used for styling ? (we can expect: Angular Material, Bootstrap, PrimeNG, Tailwind CSS) 🔎Testing: Which testi ng tools do you use? 🔎Security: Could you explain how Cross-Site Scripting (XSS) can be prev ented in Angular applications? (note: storing data, cookies, cashing) 🔎SOLID principles: What SOLID principles are you familiar with? Can you elaborate on it? (note: Single Responsibility, O pen/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion, and design patt erns are used to create maintainable and extensible code). 🔎[optional] Single-page applicatio n experience (note: good to have bcs of the performance) What can assure the seniority of an Angular Developer? ✓ Mentoring software developers in gaining experience and assuming Fro ntend developer responsibilities. ✓ Basic knowledge of tools used by Design, Backend, DevOp s teams: Design: Figma, Zeplin, AdobeXD Backend: programming languages and framewords DevOps: CI/CD pipeline and deployment (Jenkins, Firebase, GitLab, GitHub pages, Azure, AWS, GCP, etc). ✓ Senior developers prioritize testing and are skilled in working with unit tests, inte gration tests, and end-to-end tests. They promote a culture of testing within their teams. ✓ Kn owledge of advanced Angular Concepts: they understand how to optimize performance and m aintain code quality. (Hints: lazy loading, dynamic forms, dependency injection, custom decora tors, and custom directives). ✓ Knowledge of how security features work: a senior Angular de veloper should know features other than DOM only (which is a must). What are the red flags fo r an Angular Developer role at Proxify? ✖No experience with State Management tools: "I did no t have a chance to use it, but have the theoretical knowledge." ✖No Testing expertise: "I did no t write tests because it's a work of QAs" ✖Only legacy (Angular.js) or not staying updated with latest versions: "I have not tried latest versions yet, worked only with 3rd version" ✖Poor elabo rations on the tools and approaches: "I do not know if it's the best library to use, I'm just used t o it"

Summary:

Next Steps:

☐ C#

Summary:

Next Steps:

☐ PYTHON

Summary:

Next Steps:

☐ PHP

Summary:

Next Steps:

☐ FLUTTER

Summary:

Next Steps:

☐ GAME DEVELOPMENT

Summary:

Next Steps:

☐ DESIGN

Summary:

Next Steps:

☐ QA MANUAL

Summary:

Next Steps:

☐ QA AUTOMATION

Summary:

Next Steps:

☐ DATA ENGINEERING

Summary:

Next Steps:

☐ DATA SCIENCE

Summary:

Next Steps:

☐ MACHINE LEARNING

Summary:

Next Steps:

☐ BUSINESS INTELIGENCE

Summary:

Next Steps:

☐ MOBILE

Summary:

Next Steps:

☐ FRONTEND

Summary:

Next Steps:

☐ **BACKEND**

Summary:

Next Steps: