

Codility_



THE ART OF HIRING DEVELOPERS

Recruiting Strategies for Filling Front-end,
Back-end, and DevOps Roles

Introduction

Goal:

To understand and define the distinctions and qualifications between various technical roles in the company and recruit the best tech talent for each position.

Approach:

Adjust hiring approaches and interviewers' expectations based on which roles are being filled, and which hiring managers are driving the process.

Solid product development and delivery is a direct result of hiring strong technical individuals. Finding the best candidate for each position means rejecting a "one size fits all" approach. Hiring teams should instead fine-tune their hiring strategies for each of their open roles so they can build a skilled, collaborative, and motivated engineering force.

million tests on behalf of our clients. Over the years we've learned how to quickly assess programmer expertise and relative suitability for every open position our clients are looking to fill. Codility created the tech assessment movement and we've tested 3x more candidates than newer entrants in the business. The insights we've gained through this experience form the content of this ebook.

Codility works with over 1,000 businesses, and we've sent over five



Over **1,000 clients**
ranging from startups to global enterprises

Facilitating over **5 million**
candidate assessments

Different Strategies for Different Roles

We understand that there is a wide variety of tech and dev positions within each of our client's organizations. Front-end devs, back-end devs, and DevOps engineers have different day-to-day tasks, objectives, and responsibilities—so the way you hire for each role differs significantly. You're looking for different skills and specific knowledge, but it's important to also to keep candidate

expectations in mind. Of course, there is some overlap between these three roles in terms of what you're looking for in highly-qualified candidates. Any technical person you're hoping to add to your team should have a sound problem-solving approach and the ability to communicate clearly and effectively within and across teams.

Front-End Devs

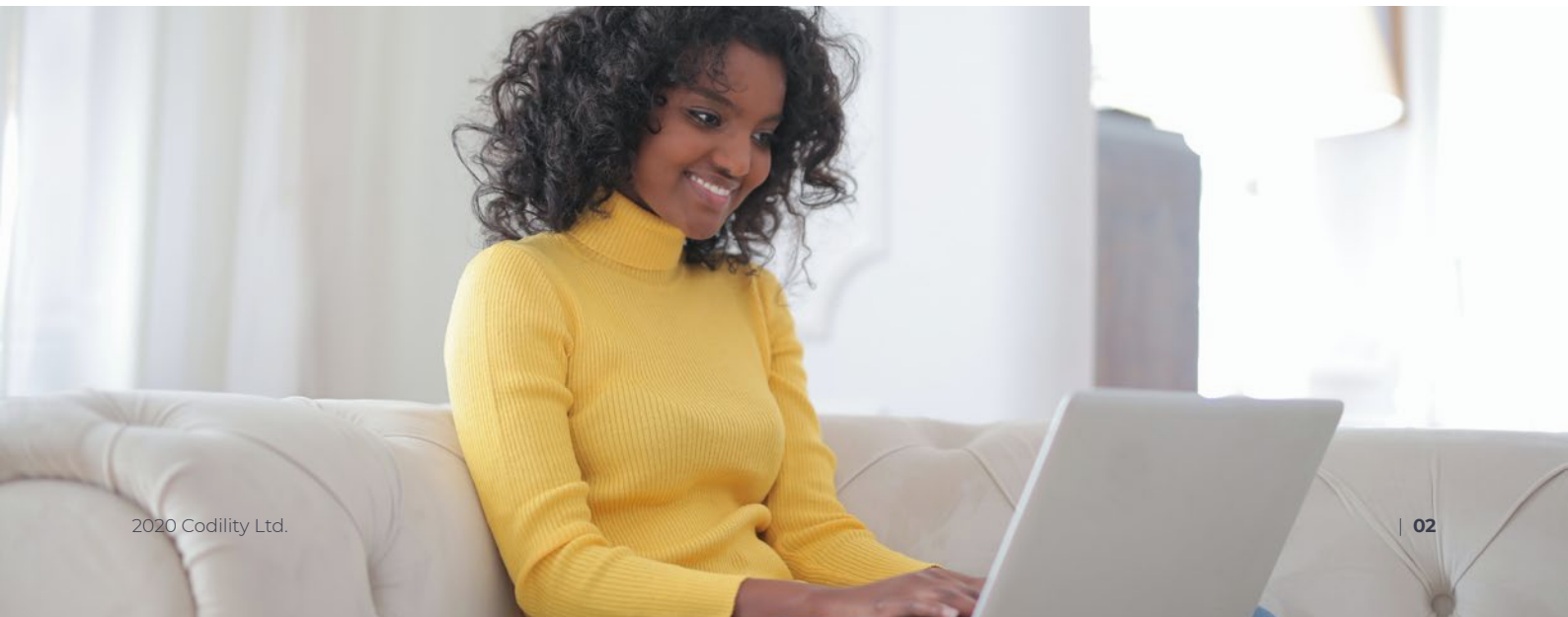
Things to look for

- Communicators
- Collaborators
- Seek Feedback
- System Thinkers
- Problem Solvers
- Fast Learners
- Detail Oriented
- User Perspective
- Builders
- Adaptable

Back-End Devs

Things to look for

- Communicators
- Collaborators
- Seek Feedback
- System Thinkers
- Problem Solvers
- Fast Learners
- Detail Oriented
- User Perspective
- Builders
- Adaptable



Front-End Developers 101

Front-end developers create user experiences and interactive interfaces for your end users.

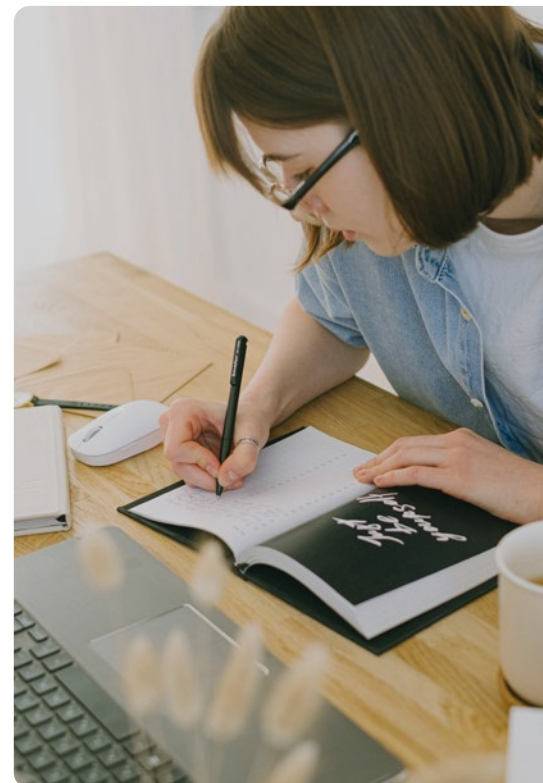
Because they work at the intersection of humans and computers, front-end developers are expected to understand both user perspectives and technical factors. For this reason, it can be difficult to find suitable candidates that not only grasp both worlds but can meld them together to produce seamless, intuitive user experiences.

Hiring Challenges

Highly talented front-end developers are in huge demand, and are very selective about the roles they take and the companies they join. Junior front-end candidates are also choosy because they often look to join an established team where they can find great coaching. For both junior and senior candidates, **make sure you keep them engaged** so they don't drop out of your process. But you'll also need to be picky about who you hire as well. The work front-end developers do touches many parts of the product and user experience, so they collaborate with a lot of teams and consequently, have many stakeholder perspectives to take into consideration. Therefore, it's essential to hire a front-end developer who plays nice with others.

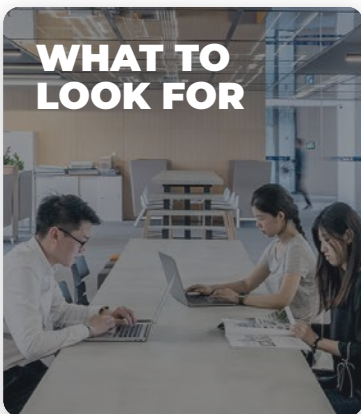
Front-end developers often don't have a formal computer science background, so they may not possess comprehensive foundational programming skills. Instead, they'll have experience with specific tools through **learning by example and**

trial by error. While they've often mastered their favorite languages, they might need to learn a lot on the job to adapt to a different stack. Few front-end developers have the know-how to create automated tests, a skill crucial for quick development.



Recruiting for Front-End

Screening

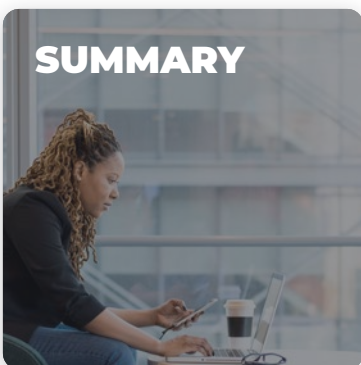


- Fundamental understanding of programming principles, frameworks, and approaches
- Familiarity with new JavaScript mechanisms
- Knowledge of closures, async/await, generators
- Stack Overflow profile, GitHub profile, personal blog, community involvement

Interview



- Ask about which languages and tools they use the most
 - Use online programming tests like Codility JavaScript-specific web tasks
 - Try language-agnostic coding and algorithmic tests
- User-orientation
 - Ability to communicate with technical and non-technical teammates
 - Acceptance of critical feedback
 - Knowledge of collaboration tools and issue trackers like Trello and Jira
- Have candidates create a small app from scratch
 - Discuss open source projects
 - Ask them how they would write code with fewer time constraints if they provided an unexpected solution
 - Gauge how they respond to feedback (again)
 - See what their day-to-day is like and get into the weeds with their workflow and work style



In screening, your priority is to find out which tools a front-end developer is comfortable using, and whether they have the aptitude and willingness to learn new languages in your stack. Learn what kind of online presence they've created over the years, and take a close look at their projects outside of work. If you plan to bring front-end candidates onsite, have them build a small app and run through it in person. Always examine a candidate's ability to translate technical concepts into fluid, beautiful user experiences.

Back-End Developers 101

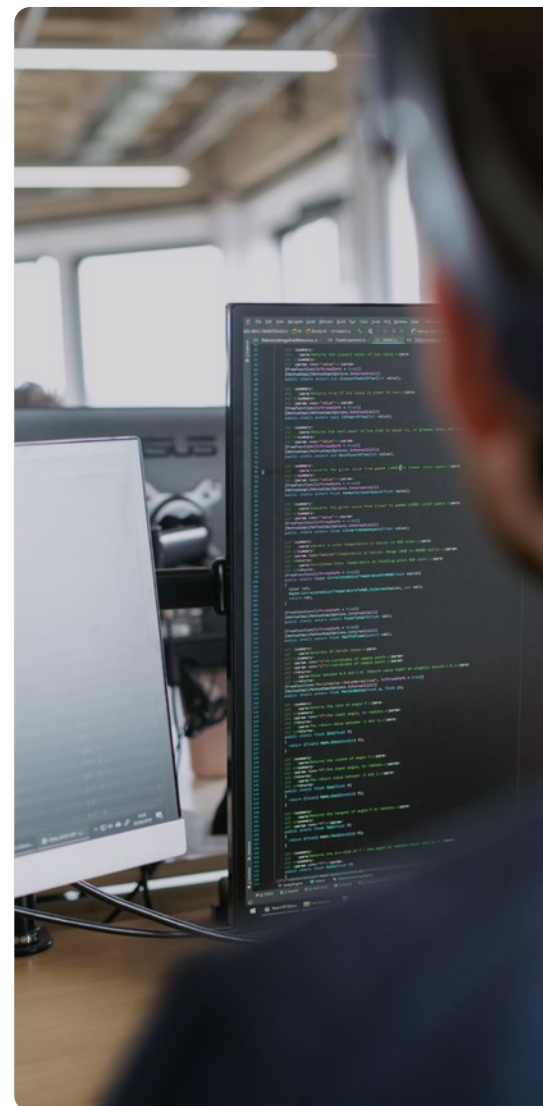
Back-end developers construct and maintain the servers, applications, and databases that enable the platform with which users interact

Because they need to translate business needs into technological requirements, back-end developers are often more technical by nature and have strong backgrounds in programming.

Hiring Challenges

Individuals who pursue a career in back-end development are usually passionate about coding and building things. These people become developers because they enjoy solving problems, creating code, and growing systems. While front-end developers might be considered user-centric, **back-end developers are often more code-centric.** When hiring programmers, it's important to gauge whether they are interested in an opportunity to work on your specific product, and towards your company mission, or if they view it as just another coding gig.

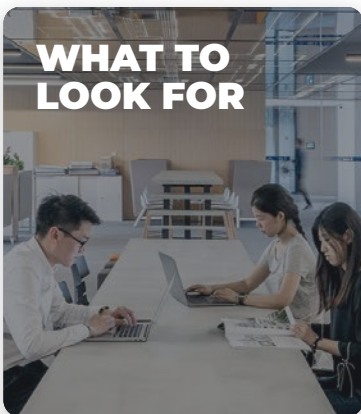
Back-end developers need to have immense knowledge of data structures and keen attention to detail to ensure things run without any hiccups. **There are a lot more back-end languages than front-end languages,** which can add complexity in finding a hire that fits with your stack. They should be armed with real-life exposure to popular bugs and data structures to be effective in the trenches.



Recruiting for Back-End

Screening

Interview



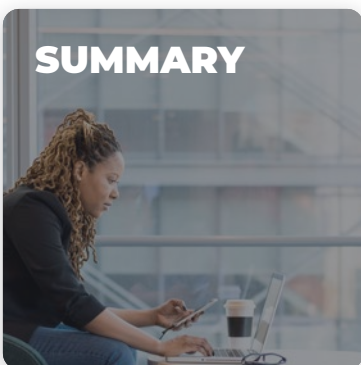
- Knowledge of algorithms and data structures
- Basic programming skills and understanding of code complexity
- Ability to learn quickly and think a problem through to find the optimal solution
- Great bug-fixing and refactoring skills

- Ability to create web apps or APIs
- Basic knowledge of how HTTP and server communication works
- Automated testing skills
- Communication skills and how they explain technical concepts - especially if the role requires collaboration with nontechnical people



- Test programming skills online with simple tasks
- Use a variety of difficult tasks in your online coding tests for senior developers
- Give candidates a logically broken piece of code and have them fix it
- Administer Codility bug-fixing and optimizing tasks
- Gauge aptitude to learn other tools and technology quickly

- See how they respond to criticism and how they support their perspective
- Suggest some code that deviates from widely accepted code styles and see whether they recognize that
- Test for obvious bugs and popular problems like N+1 problems
- Share your technical workflows and ask for recommendations to improve them



In screening, dig into how they approach problems and their ability to write code that not only works, but also scales well. **Use Codility's programming online tests to check for knowledge of algorithms, data structures, bug-fixing, and refactoring.** When back-end developer candidates come onsite, dive even deeper into their understanding of complex code, web apps, APIs, server communication, and automated testing. It's great if they are technically gifted, but use the in-person time to also find out if their interest is limited to 0's and 1's, or if they want to get behind your product and mission.

DevOps Engineers 101

DevOps engineers work at the intersection of software development, testing, and operations

They are focused on increasing the velocity of development and the delivery of applications. The most valuable DevOps Engineers identify organizational and business process opportunities, and drive change.

Hiring Challenges

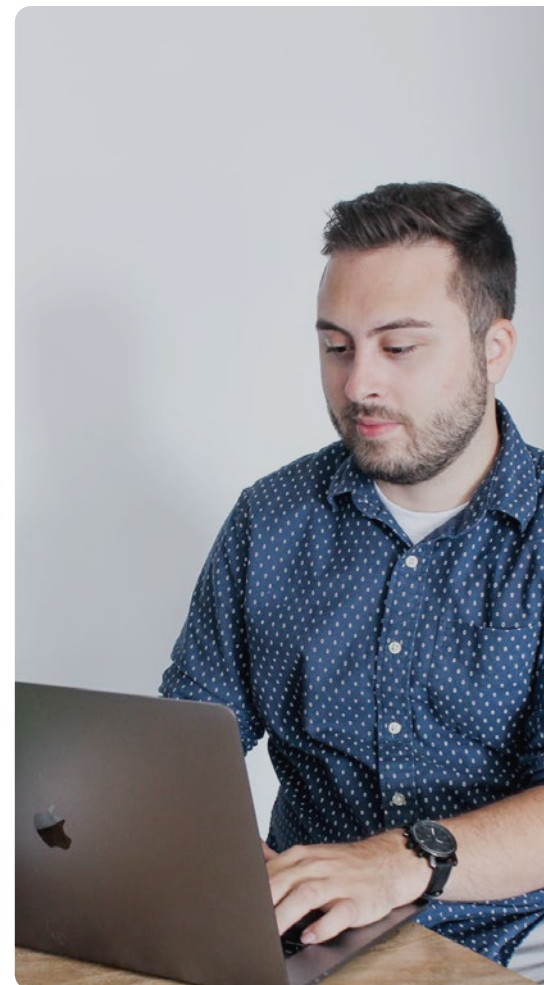
DevOps roles are, by definition, cross-functional and therefore increasingly valuable in growing organizations. This means these roles are becoming the hardest to recruit.

The role of DevOps engineer doesn't come from a single career or education track; these engineers come into the field from a variety of backgrounds. An individual might start as a software developer that oversees some aspects of operations, or they might transition from a system administrator role after learning about coding, scripting, integration, and testing.

DevOps engineers' motivations can be tricky to nail down, so it can be tough for hiring teams to pitch their DevOps roles in a way that truly resonates with top candidates.

Sometimes, DevOps engineers want to wear multiple hats because they have trouble finding a role that feels quite right. They may feel disconnected from core computer science concepts as a developer, but also might miss pure coding as a sysadmin. They want to work with a variety of languages,

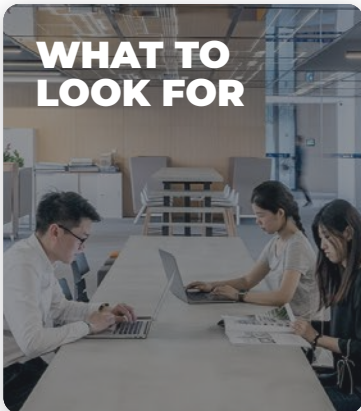
tools, and technologies to solve a variety of problems.



Recruiting for DevOps

Screening

Interview



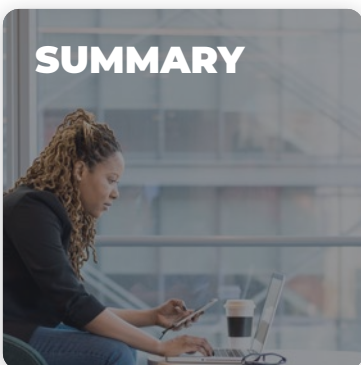
- Broad understanding of different tools and technologies
- Basic programming skills with a focus on scripting languages
- Ability to learn quickly and create high quality, long-lasting solutions
- Linux/Windows OS administration knowledge

- Ability to put teams and customers ahead of individuals
- Collaboration, flexibility, and big-picture thinking
- Knowledge of source control, continuous integration, infrastructure automation, deployment automation, orchestration, and container concepts
- Problem-solving approach (again)
- How and when they write scripts



- Look at cloud and configuration management tool knowledge
- Use Codility Multiple Choice Questions for sysadmin/Linux skills
- Administer a simple coding task related to DevOps, like learning a new tool/technology and looking at documentation to see how they approach a problem

- Give an "unsolvable" or "very hard" problem and observe their thought process
- See how they think through different scenarios, check edge cases, and ask for more data
- Don't focus on their knowledge of specific tools. Their attitude and ability to learn is more important
- Ask about projects they worked on, what technologies were used, and what they learned



In screening, see if they have solid foundations in a variety of disciplines and tools instead of honing in on expertise in any one language or technology. Get a sense of how they approach problems and how they would implement scalable solutions that last. **During in-person interviews, more closely examine what specific steps they would take to make a technical process more efficient and effective.** But remember: what's more important than their specific domain knowledge or technical skills is their approach and collaboration style. If they learn quickly, work well cross-functionally, and possess strong technical building blocks, you have a keeper.

Codility_

You'll find more guides, case studies, and data sheets to help you refactor your tech recruiting on the resources section of our website

[Go to website](#)