

Technical Hiring

Understanding the Hiring Viewpoint

Cognitive Empathy

- Using the mind to put yourself in someone else's shoes
- Job Search requires working with others

Technical Hiring Manager

- A role, not a person
 - Who has the budget and how committed and/or tenuous is it?
 - Who chooses?
 - Which candidate will be hired
 - Identify trusted resources
 - Dev leads, top performers
 - People the hiring manager wants more of
 - (confirmation bias alert)
 - Ask!
 - “What’s your role on the team?”
 - “What’s your history with the company?”
- Make sure you know who’s doing the choosing

A role, but still made up of people!

- All of them have plenty of other work to do
- They want a hireable candidate so they can get back to work
- They **don't** want your drama and psychosis about finding a job
- Help them help you...

Why are they hiring this position?

- Replacement
 - Find out what the person did
 - You'll need to be at least as good
- Growth
 - More latitude for a “team” hire
 - But more picky
- Other Reasons?

Why are they hiring (Big Picture)?

- Find the Connection between this position and the company's strategic outlook
- Understand the level of urgency

Why will a technical manager hire you?

1. Work that needs to be done today
2. Knowledge/experience the team needs today
3. Work that needs to be done tomorrow
4. (other stated lofty goals...that don't matter, see above)

Show that you can and will make contributions

1. Avoiding Red Flags
 - a. Dud
 - b. Wreak Havok
2. Where can you jump in now?
3. Where will you be able to jump in
 - a. 1 month?
 - b. 3 month?
 - c. 6 month?

Align your technical goals with their goals

Red Flags

Things that will cause a hiring manager to pass on you:

1. Appearing generally unskilled
2. Being sloppy
3. Not being able to talk about recent work in technical detail
4. Not having any recent work or projects to talk about
5. Not having technical interests (same as being interesting in “Everything”)

Appearing Unskilled

- Not understanding the basics of your first language of choice
 - Made up syntax or not valid syntax or syntax not likely to make sense:
 - `function foo[`
 - `someMethod(callback())`
- Not looking like a programmer
 - Basic control flow (looping, if/else, functions)
 - Big O in time and space
 - Basic data structures:
 - Primitives, Objects, Arrays, Functions
 - Trees, (Graphs, Sorting Algorithms)
 - Meaningful approach to small-scale problem solving

How to appear skilled

- Practice code-challenge problems
 - codewars.com
 - 30-60 minutes daily
- Value simplicity
- Want to really take it to the next level?
 - Cracking the Coding Interview: 189 Programming Questions and Solutions
by Gayle Laakmann McDowell

Will any of this help in actual job work? Debatable

Will this help in landing a job? Yes

Also check out:

<https://performancejs.com/post/hde6d32/The-Best-Frontend-JavaScript-Interview-Questions-%28written-by-a-Frontend-Engineer%29>

Get with current events and community...

- Twitter is tech heavy
- Start with people who contribute Open Source you use
- Technical body, current events: <http://esnextrnews.com/>

Don't Be Sloppy

- No excuse for bad typography
- Two faces of “engineering”
 - Running code
 - **Actual code**
- Be idiomatic
 - Read Code
 - Articles, Blogs

Not having something to say

- Based in practice, not knowledge
- Create new github projects:
 - Small: 1 x week
 - Medium: 2 x week
 - Large: 1 x month
- At least 3-4 hours/day
- Okay to have research/junk projects too
- Code along with blogs

Who would you choose?

- Chef
 - Reading and reviewing recipes
 - Cooking desserts every day
- Adult Basketball League
 - Watching past NBA finals
 - Practicing Layups and Shooting when not playing in pick-up games
- Musician
 - Reviewing music theory
 - Playing covers and working on new tunes

Do, Do, Practice, Practice

- Yes you might research, read blogs, and find new information, but tangential to your work
- What you learned:
 - Not a body of knowledge that will get you a job
 - How to learn
 - Footing and placement as a developer
 - Now the journey begins!
- Don't review, do!
- Doing software development everyday?
 - YES? You are a developer
 - NO? What are you?

Other Things You Can Be Doing

- Learn **basics** of another:
 - Language: Python, C#, Go, Java
 - Front End Framework: Vue, Ember, Angular2
- Get Involved with Open Source:
<https://medium.com/@TheLarkInn/thank-you-so-much-for-reaching-out-takes-guts-3773bac7d111>
- Give a Talk at a Meetup
 - Builds Cache/Expertise
 - Meet People who work at Companies

Claim Your Territory

Competency => Proficiency => Fluency

“Do Well” === Proficiency <-> Fluency

- What can you do well **now**?
- What do you know enough that you could do well if you spent effort?
- What do you want to know?

Plot your passion vector separately than your skills vector!

Mindset: Be a Developer, Not a Job Seeker

- Hard to do if you're not actively developing
- Remember what the hiring manager is looking for:
 - Someone to contribute to the work that needs to be done
- What they're not looking for:
 - Someone to give a chance
 - A project hire
 - High risk

Interviewing

- Approach all interviews with integrity
 - Honest and transparent
 - Be real and authentic (bacon not oatmeal!)
- Know your strategy
 - Does not mean divulging all your shortcomings!
 - Frame yourself
 - (based on claiming your territory)
- Act like a new hire

Intelligent Answers and Questions

- Don't bluster or try to snow
- Try not to speak generally
 - a. Give examples based in work you have done
- Answers based in real experience
 - a. Usually good regardless of the outcome
 - b. Having a hard time with a specific tech, or outright failing are great stories
 - c. Getting into a pros/cons discussion is a win!

When You Don't Know...

- When question is outside your realm:
 - a. Clearly define what you know well in relation to question
 - b. Ask an intelligent counter-question
 - c. Helps if you know a tidbit about what they're asking
- Example:
 - a. We use Django on the backend
 - b. My web server experience has been with node.js using express to create RESTful APIs, I haven't had a chance to use Django, how does it compare?
- Example:
 - a. Are you familiar with Ember?
 - b. My experience with component architectures has been limited to React/Redux thus far. I've heard good things about Ember, what made you decide to go with that framework?

But...

- Don't be evasive
- Feel free to say "I don't know, but it sounds interesting"
- Remember, if you start a tech conversation, it's a win!
- What would you say if asked the same thing as a new hire?

Be the New Hire

- What would you be asking if it was your first day on the job?
- Probe for the goal posts:
 - What do you see as the biggest challenge the team faces in the next 6-12 months?
 - What is the most critical thing you need from this new hire?
 - What keeps you up at night?
 - What are your goals with the company and how can this position support you?
 - What do you see as the team's biggest weaknesses right now?
 - What do you see as the team's strengths right now?
 - What are the biggest technical challenges you see the team running into?
- These questions signal self-awareness and someone empathetic to manager challenges

Probing the Dev Process 1

- Hiring: technical skills
- Reality: dev process is managers biggest challenge
- Two dimensions:
 - Project management (Delivering)
 - Engineering practices (Code Quality)

Dev Process Questions

- How often do you release?
 - Are they doing CD?
 - If not, the process has overhead
- How much time does it typically take between when dev is “done” with features and product is released?
 - Often a QA “hardening” phase
 - Restate the “qa time / (dev time + qa time)” as a percentage
- How do you interact with product management?
 - Are PM’s available and how much of the time?
- Do you reserve any capacity for development?
 - Upgrades, Pay-off technical debt

Probing the Devs

- Technical Debt
 - You can ask directly, see next question though...
 - Compare manager/dev answers
- Best developer question:
 - If product management suddenly couldn't think of any new features and you had a month to work on whatever, would would you choose?
 - (Devs will tell you all about the skeletons)
- Keep it tech:
 - Bad: What's your favorite part about working at ABC Co.?
 - The free peanuts???
 - Good: How has the transition to Ember be going?

What Just Happened?

- You just showed empathy/sympathy for managers real challenges
- You started acting like a team member in the interview
- You broke down/through the facade they were presenting (see it's not just you!)

If they feel comfortable with you technically (no red flags), and you feel like a teammate...

...you're likely to be offered the job.