
Interview This is an open source list of developer questions to ask prospective employers during the hiring process. I created this list because, frankly, the existing lists that I saw on this topic really sucked. Many of the starting questions came about from years of hiring into jobs only to find out about some lousy attribute of the company which, had I known about, I probably wouldn't have accepted. Since I began this list I've received many additional contributions from other developers.

This is not a checklist, this is not a shopping list. If you send this entire list to an employer, they probably won't be calling you back. **This list is intended to serve as a reference point for things to be aware of during your interview process.** Not all of these questions will be relevant to every person or position, you should choose the ones that are relevant to you and what you are interviewing for. It's OK for there to be questions on this list that you personally do not care about.

The questions on this list are not all appropriate for the initial interview, many could not even be answered by the HR person likely to conduct an initial interview. This list is intended to be a comprehensive collection of things a developer may want to know about a company before they accept an offer.

Use your discretion when choosing what questions to ask and when to ask them.

Contributing

Pull requests and Issues are very welcomed and encouraged, but please don't get offended if I reject your question or do not merge your question verbatim. I may ask that you reword it to be clearer, or may even rewrite it myself.

Interview This!

The Position

- Why are you hiring for this position?
- What would my role at the company be?
 - Where would I be working within the organization?
- How long am I expected to remain in this position?
- What will be my day to day responsibilities?
 - How much time do you anticipate I would spend on each one?
- What programming languages will I be expected to work in?

-
- Will the company grant me time to learn any languages that I need to use but do not know?
 - Do you have a target salary range?

Developer Coordination / Team Interactions

- How is your team structured?
 - How many developers do you currently have?
 - How large are your team groups?
 - Vertical slices or Horizontal?
- Are teams seated together?
- Do teams have isolated areas from the rest of the staff / other teams?
- How frequently do team members find themselves in meetings?
- Do your developers pair program on a regular basis?
- Do your developers use screen sharing or collaborative coding tools?
- What is your (or my future boss') leadership style?
- Do you follow an agile methodology for project management (Kanban, scrum, etc)
- How do you assign work?
 - Do you pre-assign tasks based on proficiencies, or are all members expected to be equally proficient?
- How do you estimate work?
- Single product, or will I be regularly working on different projects?
- How frequently does your company/team start a new project?
- What are your group's best and worst working relationships with other groups in the company?
- How frequently does your team interact with other teams?
- Do you find yourself frequently blocked by dependencies from other teams?
- What hours does the team work?

Development Process

- What source control do you use? Can you explain why you chose it?

-
- Are your repos hosted in-house or on a third-party service?
 - If in-house, do you use a repo management application such as GitHub Enterprise or Gitlab?
 - What is your workflow currently, with regards to developers pushing changes.
 - Do you do pull requests, or does everyone just merge to a central repo?
 - What branching method do you use? (Git-Flow, Github-Flow, Environment Merging, something custom)
 - Are you using a ticket system or is it more play it by ear?
 - Do you use the same system for both bugs and new features?
 - How is priority determined?
 - What comes first, bugs or features?
 - Are detailed requirements for tasks determined and documented ahead of time?
 - How are tasks added to the backlog?
 - Who determines the features that would go into the product? How are these chosen?
 - How are developers involved in that decision?
 - How are testers involved in that process?
 - How do you track development time?
 - Do you have a code review process?
 - Does your code review process prevent deploying code?
 - Does your code review process promote empathy?
 - Does your team encourage the use of SOLID and DRY design principles to avoid cyclomatic complexity?
 - What is your take on object calisthenics?
 - Do you have established code style rules?
 - Did you create your own style guide, or are you using a third party's (PEP8, PSRs, Standard JS, etc)
 - Is there an automated linting process to validate your styles?
 - Tabs or spaces? (If relevant)
 - Allman or BSD braces? (If relevant)
 - Semicolons? (If relevant)

-
- What are your development environments like?
 - Virtual Machines? Local (VirtualBox) or Remote (ESXi)?
 - Does everyone have an identical development environment?
 - Are you using vagrant and/or puppet/chef?
 - How closely do the dev environments mirror your production environment?
 - Will I be provided with a new laptop?

(Author Note: These are basically more direct versions of the Joel Test “best tools” question.)

- Windows, Mac or Linux? Do I get a choice?
 - Am I allowed to install anything I want on that laptop?
 - Will it have an SSD and as much ram as can fit?
 - How hard do I have to justify software purchases?
 - How often will I receive hardware upgrades?
- What is your release schedule like?
- Will I communicate directly with clients on a regular basis, or does this typically happen through an intermediary?
- What can I expect to see in terms of project specifications and/or mock-ups prior to beginning a new project?
- How is QA/testing done?
 - Are developers expected to do testing as well? To what extent?
 - If there is a separate testing team, how is the co-ordination done between testing and development?
 - How is the decision made that a release is ready?
- Who designs the User Interface?
 - Are developers expected to do it?
 - Are there graphics designers in the company?
 - Are there UX designers in the company?
- Are there compliance requirements in the company like CMM or SOC?
- Do you have an SLA (Service Level Agreement)?
 - Do you guarantee any of the following? If so, how long?
 - ★ Turn Around Time (TAT)?
 - ★ Average Speed to Answer (ASA)

-
- ★ Time Service Factor (TSF)
 - What is the escalation plan? What are the consequences if the plan is not followed?
 - Do you have on call hours?
 - ★ What is the on-call schedule?
 - How satisfied are your engineers with their current toolset? If they had to replace one tool, which would it be and what would they replace it with?

Monitoring / On-call duty

- Do you have any application-level logging? If so, how are logs accessed?
 - Are logs aggregated across multiple hardware?
 - Are your logs search-able?
 - How verbose is your logging?
- Do you use any monitoring software? (Nagios, Icinga, Zabbix, etc.)
- Do you regularly record and review application performance metrics? How are performance optimizations prioritized with respect to other types of tasks?
- How does it inform staff of error conditions? (PagerDuty, Email, SMS, big monitors in each teams room, etc.)
- Is there a permanent on-call duty for each IT team?
- Is there a permanent “control center” that keeps track of events and informs the responsible on-call duty?
- Is there an escalation process if someone cannot be reached?
- Do the developers also have an on-call duty?
- Does time working on incidents/problems after-hours, when on on-call duty, count as overtime?
- Are employees expected to be doing after-hours work while waiting on-call?

Remote

- What timezone/hours does the company work in?
- Will the company pay for home office equipment such as electronics or furniture?
 - If yes, will that equipment be considered company property?
 - Would I have to return it if/when I leave the company?

-
- Do you have a team chat setup such as Slack or Jabber? Do your developers actively use it, is it their primary communication channel?
 - Do your developers use video chat software such as Skype or Google Hangouts?
 - If a portion of the team works in-office, do you have a dedicated computer to be used for video chat with remote employees?
 - Will I have to work over a VPN?
 - How frequently will I be expected to visit the office?
 - Will my visits and travel costs be reimbursed, or covered outright by the company?
 - How flexible are my hours? Can I take time off during the day if needed and make up for it in the evenings?

Open Source

- Do you use open source libraries?
 - Are you aware of the licensing on those libraries?
- Does your company release open source code?
- Does your company contribute to (or encourage their developers to contribute to) open source libraries?
- What is the company policy with regards to me releasing open source code (personal projects)?
 - If there is an approval process, how lengthy is it?
 - Is the process formalized?

Codebase / Architecture

- How old is your codebase?
- Do you have an automated test suite?
 - What libraries and tools do you use?
 - What sorts of tests do you use? (unit, integration, system, load, ...)
 - What is your testing methodology? (BDD, TDD, Spike & Stabilize, ...)
 - What is your current level of test coverage? Are you happy with it?
- Do you regularly correct technical debt?

-
- On a scale of 1 to 10, how much spaghetti code do you have?
 - How well-documented is your codebase?
 - Do you use automated documentation systems like PHPDoc or JSDoc?
 - Do you maintain a wiki?
 - Pure CSS, or compiled middleware (LESS, SASS, etc)?
 - What browser and operating system versions do you support?
 - Does your codebase require a build process, and is it automated?
 - Do you have a continuous integration process implemented?
 - Do you use MVC or similar code structuring?
 - Is there an in-house framework, and if so who controls it?
 - Do you host your product yourself (Local, CoLo, VPS) or is it running on a cloud platform such as AWS or Heroku?

Diversity

- What percentage of the company is non-male?
- What percentage of the company is non-white?
- What percentage of the company is LGBTQ?
- What percentage of the company is non-development staff?

Culture

- What is the rhythm to the work around here? Is there a time of year that it's all hands on deck and we're pulling all-nighters, or is it pretty consistent throughout the year? How about during the week / month? Is it pretty evenly spread throughout the week / month, or are there crunch days?
- What made you (the interviewer) choose to join this company?
 - What do you enjoy the most about working here?
- Who are the heroes at your company?
 - What characteristics do the people who are most celebrated have in common with each other?

-
- Is there a company reward system for employee accomplishments?
 - What type of people are successful here? What type of people are not?
 - Am I allowed or expected to take my work home with me?
 - What are the expectations with regards to hours worked, deadlines, and overtime?
 - How much vacation time do you provide?
 - How much lead time is expected on vacation requests?
 - Open office, personal offices or cubicles?
 - Is there a dress code?
 - What relationship does your dev department have with your sales department? Who sets the deadlines?
 - Does the company provide snacks and/or drinks?
 - What are your expectations for how many productive hours a developer will have per day?

Company

- Is your company currently profitable?
- What's the biggest change your group has gone through in the last year?
- If I get the job, how do I earn a "gold star" on my performance review? What are the key accomplishments you'd like to see in this role over the next year?
- Which competitor are you most worried about?
- How does sales / operations / technology / marketing / finance work around here? (I.e., groups other than the one you're looking to work in.)
- What's one thing that's key to this company's success that somebody from outside the company wouldn't know about?
- How did you get your start in this industry? Why do you stay?
- What keeps you up at night? What's your biggest worry these days?
- What is your biggest complaint about the company?
- If we have a very successful year, what would that look like? What will have happened over the next 12 months? How does this position help achieve that?

-
- How does the company / my future boss do performance reviews? How do I make the most of the performance review process to ensure that I'm doing the best I can for the company?
 - What information is shared with the employees (revenues, costs, operating metrics)?
 - Is this an open-book shop, or do you play it closer to the vest?
 - How is information shared?
 - How do I get access to the information I need to be successful in this job?
 - Who is your health-care provider?
 - What percentage of insurance does your company pay?
 - Does your company provide maternal/parental leave?
 - Will the company pay for training programs / certifications / conferences? What is the approval process like?
 - What is the company policy with regards to side projects? Am I allowed to work on my own sites?
 - Do I own the code I make in my own time on my own hardware, or does the company claim it as theirs?

PHP

- Do you use a public framework or is it an in-house environment?
 - When you find a bug in a public framework, do you give it back to the community?
- Do you use PHP-driven HTML templates, or a third-party template engine such as Smarty or Twig?
- Do you use Composer?
- Do you encourage your developers to take the ZCE exam?
- Which version of PHP are you using?
 - What is the update plan for new PHP releases?

JavaScript

- What is your frontend software stack? (jQuery?, Underscore/Lodash?, Angular/Ember/React?, etc)
 - Why did you make those choices?

-
- Is your front-end code bundled using a module loader such as Webpack or Browserify?
 - Do you use a templating engine, such as EJS, Jade, or Handlebars?
 - Do you use a compiled language such as Coffeescript or Typescript? (If yes, is it required?)
 - Do you use NodeJS as a software platform? (ie, beyond tooling)

System & Network Administration / IT Operations

- Do you use a configuration management tool? (Puppet, Chef, cfengine, Ansible)
 - Why was it chosen?
 - Is its use accepted throughout your IT staff?
- Are configurations version controlled?
- What is the process for granting a user access rights (RDP, SSH, etc.) to a system?
- Are there multiple access levels for different classes of user?
- Do developers have admin/root rights on systems?
 - If yes: Why?
- Do you have different staging environments for testing/development? (Like: DEV, QA, PreLIVE, LIVE)
- Are developers allowed to connect to systems outside of the development environment?
- Do you have a Change Management process? (ITIL, etc.)
- How do you organize system administration, application development, application deployment and application operating so they fit together?
- Is there a wiki for server documentation/howtos/best practises?
- Do you use the same OS distribution on all your servers, or is each server configured for specific needs?
 - Why did you choose your OS? What were the requirements?
- Are development systems and services standardized, or do developers choose their own environments?
- Do tools need to be approved before use, or may I use whatever I want?
- How frequently do you replace server hardware?

-
- Do I have to replace hardware parts myself or is there a dedicated team / external contractor?
 - What software / services do you use to load balance?
 - Are your applications architected for horizontal or vertical scaling?
 - What is the average uptime of your servers?
 - Do you consider uptime to be a good indicator for system reliability?
 - How do you test fault tolerance? Do you have some kind of “Chaos Monkey”?
 - Is there a process for self-build packages (.deb/.rpm/.msi) to be put on some internal repository when an official repository can’t provide a package/bugfix?
 - How do you manage IP addresses and DNS records on your network?
 - Do you have plans for (switching to) IPv6?
 - Do you categorize your networks? (database server network, frontend network, middleware network?) or is everything mixed together in various networks?
 - Are DEV/QA/PreLIVE/LIVE systems all in one big network, or is each on a separate network? Are they firewalled so a DEV system can’t DoS a LIVE system?
 - What is the process for managing rules on internal/external firewalls?