stripe

On-site interviews for Engineering: What to expect

We're excited you're coming to interview at Stripe! We want your interviews to be as comfortable as possible, and for you to feel like you've presented yourself well. To help you out, this guide explains what to expect and how to prepare.

TECHNICAL INTERVIEWS

Our technical interviews simulate the engineering work you'd do day-to-day at Stripe.

In these interviews, you'll consider technical problems and then write working programs that solve them. In some cases you'll also have a design discussion with your interviewer before diving into coding.

You should use whatever programming language, environment, and editor you're most comfortable with. We encourage you to use online resources (docs, Stack Overflow, etc.) as needed during the interview.

Consider your interviewer a collaborator on the problem. Share what you're thinking with them and ask them for help when needed.

How to prepare

- You should be ready to write, run, and debug code in your language of choice. If you have a laptop you're comfortable using, feel free to bring that. We can also provide you with a computer for interviewing if you'd like; just let us know in advance.
- Be prepared to run a standalone program in your environment, print output, include common libraries, use debugging tools, and use built-in data structures..
- It can be useful to set up common project boilerplate before the interview if appropriate for your language. Be ready to run a simple *Hello, World!* program.
- One of the interview questions will be structured around HTTP requests. We're not looking for detailed knowledge, just a basic understanding of what a request and response look like. If you're not familiar with the HTTP protocol, this article is a good reference.
- Come with questions for your interviewer. These can be technical or not. Think about what's important to you, and ask us how well Stripe does on those things. Don't be afraid to ask hard questions. Julia Evans put together a great blog post on questions for interviewers.

stripe

ENVIRONMENT SETUP

If you're interviewing in one of the languages below, we have some specific setup instructions you can use to prepare in advance.

- *Ruby*
  Make sure you have some version of Ruby installed on your laptop. You'll also need to have Bundler installed to help manage dependencies of some of the code we'll provide. You can install it by running `gem install bundler`, if you haven't already.
- *Java*
  We've created a Maven project to help you determine if your computer is configured correctly to write Java, at least at a level that some of our questions expect. Before your interview, please visit the repository and follow the commands under the "Getting Ready" section.
- *Python*
  We've created a Python project to help you determine if your computer is configured correctly to write Python for the purposes of our interview.
- *Scala*
  We've created an SBT project to help you determine if your computer is configured correctly to write Scala. Before your interview, please visit our repository.
- *Other*
  Make sure you can compile and run a simple *Hello, World!* application, including Makefiles, POMS, or Gemfiles, etc.

HOW YOU'RE EVALUATED

Here are some things we look for in technical interviews:

- *Problem solving*
  How effective are you at understanding the problem and devising a solution for it?
- *Design*
  How do you design and lay out your code? Is it well organized with easy to understand interfaces?
- *Correctness*
  Do you think clearly about the correctness of your code? Do you use explicit reasoning and/or tests to check correctness? Do you consider edge and error cases?
- *Debugging*
  Do you spot the bugs in your code? When you find one, can you fix it?
- *Programming language familiarity*
  Are you comfortable with the language you're writing? Is your code idiomatic for the language?

stripe

- *Tools familiarity*
  Are you set up to write code? Can you run it? You don't need to have an advanced editor or setup; you should just be comfortable and productive in your chosen environment.
- *Navigating codebases*
  For one interview, we'll have you fix a bug in an open source project. Can you navigate an unfamiliar codebase and figure out what's going on?
- *Communication*
  Do you clearly explain your thoughts? How well do you respond to feedback or suggestions from your interviewer? Do you seek help or guidance when stuck?
- We used to apply a 'Sunday Test': "If this person were alone at the office on a Sunday, would that make us want to come in just to hang out with them?". However, this was the wrong phrasing for the idea we really cared about (we don't encourage people to work Sundays, and we want people who would make excellent coworkers rather than be our personal friends). Instead, we now ask: "Is this someone we'd actively seek to work with"?

NON-TECHNICAL INTERVIEWS

These interviews give us a chance to learn more about you, your goals, and your interests. It's also a chance for you to learn about Stripe and hear answers to more of your questions.

How to prepare

- Come with questions for your interviewers. Consider what's important to you in a job, what you'd like to see in a work environment and colleagues, and what things you'd like to avoid. Feel free to ask about e.g. company goals, team organization, Stripe's values, etc.
- You may check out Stripe's website and recent news to see if anything is particularly interesting to you. You won't be quizzed on anything here, but it could be a good basis for discussion.

Thank you again for taking the time to come in to Stripe. We want to make your experience as pleasant as possible, so please let your recruiter know if there is anything else we can do to set you up for success. Our team is very excited to meet you!