

Engineer Team Screen Interview Information

We're excited you'll be interviewing at Stripe! This interview will evaluate your ability to solve a programming exercise in a readable way, as well as your knowledge in automated testing. We don't use Leetcode or trivia, or ask trick questions. This exercise is an example of the kind of coding we do regularly in our daily life at Stripe. In fact, every part of our interview process is designed to give you streamlined examples of the kinds of real-world problems we solve day-to-day.

Overview

- The technical portion of the interview should be approximately 45 minutes, with 15 minutes reserved for introductions, discussions, and asking questions.
- Team screens are done over Zoom — the link will be in your calendar event. We recommend using the programming language you're most comfortable with (any language is ok.)
- Your goal is to write clean code that works. Focus on correctness, readability, and testing over performance, and pick whatever approach you think you can get working the quickest. We won't ask you to find the most efficient possible solution to a problem, but we might ask you to explain if your code could be written in a better way.
- As a heads up, there might be an additional engineer in the meeting. They will be shadowing as part of our interview training program.
- **Please note that the use of any AI tools (e.g., Copilot, Gemini, TabNine, ChatGPT, LLMs, or any other AI assistants) in solving interview problems is strictly prohibited. This includes the use of AI results in search tools, such as Gemini for Google Search.**

FULL STACK, BACKEND & MOBILE INTERVIEWS

- You should be familiar with common operations on arrays, maps, strings, string parsing, and integers in your language of choice, as well as your ability to write solid code and debug effectively. Having familiarity with your favorite language's standard library and built-in functions will also help you to be successful.

- Note: Some candidates have found that using dynamically typed languages is more efficient for these interviews. However, it is possible to be successful using any programming language, and we don't recommend using a programming language that you're not familiar with.
- You may choose whether to use HackerRank, an interactive IDE, or your own IDE setup. If you choose HackerRank, at the start of the interview, your interviewer will share a technical problem using a link from HackerRank, an interactive IDE. You can try out using the HackerRank IDE in [their sandbox](#). If you decide to use your own IDE setup, we'll ask that you use Zoom to share your screen with your interviewer. If for any reason you'd prefer not to share your screen, or if your preferred tools aren't available on the computer you'll be calling from, we've found that HackerRank can be a great substitute for screen-sharing. If you run into issues using either Zoom or HackerRank, CodeBunk will work as well.

FRONTEND INTERVIEW (only for those who applied to a role titled "Frontend Engineer")

- During your interview, you will be asked to build a small, self-contained JavaScript component, the kind you might find in a component library. We'll be looking at how you design, structure and debug your code.
- The task that you'll be completing in the interview may involve working with the DOM, forms, and manipulating data (strings, arrays, objects) in JavaScript. You'll be able to search the web for documentation if you need it, and you're welcome to use utility libraries (lodash, jQuery, etc.) or UI frameworks (React, Angular) as you see fit.
- We recommend that you spend some time before the interview setting up your development environment so that you can dive straight into code. This might mean setting up a plain HTML page with your preferred libraries like jQuery included. If you're using React, set up a basic app with any transpiling or bundling you might need (create-react-app is great for this). We want you to be able to maximize the time allotted for this interview. If you have questions about tools, please let me know.
- For this interview, we'll ask that you share your screen over Zoom [0] with your interviewer. You'll have a chance to work through a coding exercise using the tools of your choice, so please make sure you have anything you need to run your code installed before starting the interview.

General Tips

HOW TO PREPARE

- Be well-rested and ready to think on your feet, as the interview will require you to problem-solve in real time.
- For the sake of time during the interview, please make sure you have your preferred environment for writing and executing code set up beforehand.
- Come prepared to do comprehensive, programmatic checking of your code. You can write your own tests manually. You don't have to use a testing library.
- If you are using an unfamiliar IDE or runtime environment, take some time prior to the interview to acquaint yourself with how to run and debug code.

PRACTICE

- ***Our interviews focus less on data structures and algorithms, and more on practical problems like what you'd encounter in your daily life at Stripe.*** That said, we know that turning ideas into code in real time can be uniquely challenging! We find that candidates who practice before their interview feel more prepared.
- We recommend practicing timed coding problems. Focus on writing code quickly, ensuring correctness, and explaining your thought process out loud as you work.
- While our interview problems are different from exercises on [LeetCode](#), [Interview Cake](#), and [Interviewing.io](#), some Stripes have found these sites helpful for practicing timed coding challenges.

DURING THE INTERVIEW

- Feel free to reference documentation, Google, StackOverflow, or any other resources, just as you would at work or for a school assignment.
- Use your language's standard library or 3rd party code as much as you want, if you think it would be helpful.

Keep the following questions in mind:

- Do you understand the problem? Some of our questions are longer and more detailed. Read the instructions carefully and ask clarifying questions. We encourage asking clarifying questions and have found that it confirms mutual understanding and sets the stage for the remainder of the interview. We'd rather you ask questions than dive straight into the problem.
- Did you come up with a good design to answer the question? Don't forget to communicate your design decisions to your interviewer. This is not a system design interview, but we still like to hear the design decisions you're making!
- Did you proactively and effectively test your own code? Make sure you're covering corner cases and edge cases.
- Is the code correct? Did the code result in the correct output? Did it return an error when it should?
- Does the code follow typical language-specific conventions? Is it easily readable?
- Are you systematically debugging and finding errors? Be methodical. Identify errors on your own and handle them.
- Did you communicate clearly and share your thought process? Talk out loud and explain why you're doing what you're doing as you code!

What We Evaluate

We want you to have the opportunity to present your best self. If you're considering what to emphasize, these are our top priorities:

- **Speed and efficiency**
 - Can you make progress on a multi-part problem within the allotted time?
- **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?
- **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.
- **Communication**
 - Do you clearly explain your thoughts as you work? How well do you respond to feedback or suggestions from your interviewer?

Sample Scenario

PROGRAMMING EXERCISE

You are encouraged to ask the interviewer questions as you read and after reading.

At Stripe we keep track of where the money is and move money between bank accounts to make sure their balances are not below some threshold. This is for operational and regulatory reasons, e.g. we should have enough funds to pay out to our users, and we are legally required to separate our users' funds from our own. This interview question is a simplified version of a real-world problem we have here.

Let's say there are at most 500 bank accounts, some of their balances are above 100 and some are below. How do you move money between them so that they all have at least 100?

Just to be clear we are not looking for the optimal solution, but a working one.

None

Example input:

```
AU: 80
US: 140
MX: 110
SG: 120
FR: 70
```

Output:

```
from: US, to: AU, amount: 20
from: US, to: FR, amount: 20
from: MX, to: FR, amount: 10
```

Potential follow ups/parts (in no specific order):

1. **(Practical)** If this code will be used to move millions of dollars in production, how would you change it? Specifically, we just eyeballed that our end goal of each balance ≥ 100 is met, how would you check that in reality? What should we do if the check fails?
2. **(Algorithmic)** Do it in the minimum number of moves. For the input data in the original prompt:

None

```
from: US, to: FR, amount: 30
from: SG, to: AU, amount: 20
```

What We Evaluate For This Interview

- Correctness/Testing/Debugging
- Interaction and collaboration
- Thought process and abstractions
- Writing code

Frequently Asked Questions

Wondering what to wear?

Many people at Stripe show up to work in jeans and a t-shirt (or button down/blouse). We recommend simply wearing what you're most accustomed to wearing, and we promise no one will judge you for it!

Why is there more than one Stripe in my interview?

You may notice that many of your interviews will have more than one Stripe team member on your Zoom. This is to ensure we are holding a consistent hiring bar, and to make sure we have enough fully trained interviewers to help us scale. The additional folks on the Zoom are shadowing the interview, which is part of our interviewer training process. You will work directly with your interviewer, who is fully trained, while another Stripe will observe the process to make sure they can replicate it fairly and consistently.

Accommodations

Stripe is committed to providing equal employment opportunities for all candidates. If you require any reasonable accommodation for your interviews, please reach out to accommodations@stripe.com. Your message will be confidentially received by a member of our people team who will assist with your request. Please share only relevant information needed for your accommodation (not including confidential health or medical information).

Thank You

Thanks for your interest in Stripe! We appreciate the time and energy you've already committed and for giving us the opportunity to consider you. 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!

Refer to your role's job description to prepare for your interview by visiting [Stripe's job page](#).

