

Video interviews: What to expect

We're excited you'll be interviewing at Stripe! Your video calls will give you an opportunity to meet some more of our team and get a feel for what it's like to work here. We've put together this short guide to give you an idea of what your day will involve. Like many things at a fast-growing startup, we're constantly improving and iterating on how we interview, so if you have any comments, concerns, or questions, please let us know. We look forward to meeting you soon.

Some general tips

Many of the interviews included during your onsite visit are designed to help us understand how you might approach common scenarios encountered at Stripe. Most of the exercises you'll do in these interviews will be directly related to Stripe. However, you're not being tested for your knowledge of Stripe and you'll always have the opportunity to ask us questions about anything you don't understand.

HOW TO PREPARE

- Take a look at our [blog](#) and [Twitter account](#).
- Do some background reading about our products and documentation. This will provide some context for the questions we'll ask in the interviews.
- Be well-rested and ready to think on your feet, as some of the interviews will require you to problem-solve in real time.



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. However, please do not just choose a language that is good for algorithm demonstration, you will actually be working with real large software. We encourage you to use online resources (docs, Stack Overflow, etc.) as needed during the interview.

Additionally, if you have a Mac OS/Linux machine available, we recommend using that however it is **not necessary** for you to use one of these in order to perform the interviews effectively.

Consider your interviewer a collaborator on the problem. We suggest "working out loud" with your interviewer and asking any questions that you have as they come up, for example about constraints on the system.

HOW TO PREPARE

- You should be ready to write, run, and debug code in your language of choice. If you have a personal (not work-issued) 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 a 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 may 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](#).

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. Some of the languages below contain tests, in addition to build instructions, to verify the environment setup is working. Please ensure you are able to run these tests prior to your interview.

- **Ruby**
 - Ensure you have a stable and supported version of Ruby installed on your computer. See the currently stable versions of Ruby [here](#). 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. Additionally, some dependencies may require a C compiler to build. Ensure that you have `make` and `cmake` installed. Even better, if you are using a Mac and have `homebrew` installed, you will be in good shape.
- **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. Note, we will be working with Python 3. Additionally, some dependencies may require a C compiler to build. Ensure that you have `make` and `cmake` installed. Even better, if you are using a Mac and have `homebrew` installed, you will be in good shape.
- **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](#).



- **C#**
 - We've created a [C# Solution](#) to help you determine if your computer is configured correctly to write C#, at least at a level that some of our questions expect. Before your interview, please visit the repository and follow the instructions under the ["Getting Ready" section](#).
- **C++**
 - We've created a [C++ project](#) to help you determine if your computer has the required softwares and is configured correctly to write C++, at least at a level that some of our questions expected. Before your interview, please visit the repository and follow the instructions under the ["Getting Ready" section](#).
- **Javascript**
 - Make sure you have [Node.js](#) and [NPM](#) installed and make sure you can run a simple Hello, World! application.
- **Other**
 - Make sure you can compile and run a simple Hello, World! application, using the relevant dependency management system for your language.

Design Challenge

We also want to better understand how you design systems, and organize and communicate your technical thoughts.

For this interview you'll design a technical system using an online collaboration tool. A Stripe engineer will share the context and prompt and you'll collaborate on a solution, preferably using Whimsical or Zoom Whiteboard (instructions for each below). Consider your interviewer a collaborator on the problem and plan to work with them using your preferred collaboration tool. We suggest "working out loud" with your interviewer and asking any questions that you have as they come up, for example about constraints on the system.

HOW TO PREPARE

Since you will be completing this interview over VC instead of in-person, you can share your design with the interviewer using one of the following tools:

- Whimsical collaboration tool (preferred)



- Zoom's white board tool
- Any preferred drawing tool that is installed on your device, a tablet, or even a whiteboard if you have one handy at home

Feel free to use this [resource](#) to familiarize yourself with and get setup using Whimsical or Zoom's collaboration tool.

Bug Squash

This interview evaluates your ability to understand and navigate an unfamiliar, but well-written, idiomatic code base. You'll navigate a large codebase to spot (and fix!) bugs in the code.

WHAT TO EXPECT

This is something that you probably do on a regular basis. At the beginning of the interview you'll be given a fairly large and complex Git repository to download along with some associated tests. Run the tests with the interviewer, and of course these will fail because of the bug(s). Work with the interviewer to make your way through the codebase, find the location of the bug(s), write a quick fix and run those tests again. Hopefully everything will pass and that will mean you completed the question.

HOW TO PREPARE

We are much more interested in your strategy and approach to debugging rather than aimlessly looking for bugs:

- Sometimes if people don't have a strategy they might stumble around the code
 - It is not a sign of success if you happen to find the bug on accident without a clear strategy
- Make sure you have a methodical approach to debugging with stating a hypothesis and validating or not validating it
- We like it even more when you understand what the code base is and show that you understand what the code is doing
- How many bugs? I can't tell you that (you're not finding a million little bugs)



WHAT WE EVALUATE FOR THIS INTERVIEW

- Finding bugs
- Forward progress
- Interaction and collaboration
- Linguistic knowledge and tooling

Programming Exercise

This interview is an in-depth programming project so we can get a sense of how you tackle problems and communicate with other engineers.

WHAT TO EXPECT

The format of this interview should loosely mirror the technical assessment you have already passed. View this interview as a paired programming problem where communication and collaboration are incredibly important.

HOW TO PREPARE

This will be quite a long question and a lot of the time, candidates will not finish. Don't worry, completing this problem is not a measure of success for us here. We really emphasize your approach to solving the problem as well as the code and abstraction quality that you are writing.

- On a sliding scale of speed vs quality, go with quality as long as you are still able to make forward progress
- Make sure you spend a little bit of time up front designing your solution so you have a sense of if the code will work before you begin writing idiomatic code
- We do care about how thoughtful you are for code cleanliness and vocalizing trade-offs
- We do not know what you are thinking, and would love to hear your thought process



WHAT WE EVALUATE FOR THIS INTERVIEW

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

Integration

This interview evaluates your ability to write code within an existing system, familiarity with libraries, and your aptitude at combining existing code to fulfill a spec.

WHAT TO EXPECT

There will be a spec provided to you at the beginning of the interview and we *highly recommend* reading it through completely before you begin so you fully understand the requirements. When you are ready to write your implementation, try to think about test cases that you want to consider. If you are short on time that is okay, but we recommend talking through the edge cases that you want to consider and what may be wrong.

HOW TO PREPARE

Testing is a very important factor in this interview because you are introducing new code or you are manipulating existing code, and we want to make sure you are not breaking something in the overall system just to make a small change.

WHAT WE EVALUATE FOR THIS INTERVIEW

- Ability to help yourself
- Abstractions and writing code
- Correctness/Testing/Debugging
- Interaction and collaboration

Hiring Manager Chat

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



to more of your questions.

HOW TO PREPARE

- Consider what's important to you in a job, what you'd like to see in a work environment, how you partner with colleagues and managers, and what you'd like to avoid. Feel free to ask about company goals, team organization, Stripe's values, etc.
- Take some time to review the job description and think of how your past experience relates to this role. You might find it useful to come up with a few flexible examples of projects you've worked on in the past. Examples of past experience and work often come up during interview conversations, and often it's easiest to draw upon a few vetted examples instead of coming up with something on the spot! (Please don't discuss anything that your previous or current employers would consider confidential.)
- Sample topics to come prepared to speak about:
 - Describing a technically challenging project and how you approached it
 - Technical contribution that allowed others to be successful
 - Project that you worked on that were very cross functional
- You may want to check out [our website](#) to see if there is anything particularly interesting to you that you'd like to chat about during the interviews.

WHAT WE EVALUATE FOR THIS INTERVIEW

- Career fit
- Communications
- Work experience

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 nine priorities:

- **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.
 - **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?
 - **Stripe fit**
 - Are you actively inclusive and supportive of individuals who you work with? Will you fight against other competing forces on behalf of our users? Do you take pride in what you've done, but focus on what you can improve? Do you move with urgency and focus, without thinking the grunt work is beneath you? Do you reason from first principles to get to the right conclusion?
-



How are video interviews different than in-person interviews? Is there anything special I should know?

We understand that interviewing virtually is a different experience than interviewing in person. While some candidates may prefer this format, others may have preferred in-person interviews. We appreciate your flexibility with this!

Your interviews will use our video conferencing tool, Zoom (a service very similar to Skype). There are a few things you should know about interviewing over Zoom:

- Unless otherwise indicated by your recruiting coordinator, all of your interviews will be hosted on the same Zoom Meeting ID, which was provided in your interview confirmation email. You do not need to log off and log back on in between your interviews – simply stay on the line and wait for the next interviewer to arrive. If you have any technical difficulties the day of your interview, you can reach out to your recruiting coordinator via the cell phone number provided in your confirmation email.

How long will the interview take?

While some candidates may have a shorter day with us, it's best to plan to be at Stripe for about five hours. If you have any time constraints, please be sure to let your recruiter or recruiting coordinator know.

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.

—

Additional notes and accommodations

Breaks:

- We will do our best to provide a 15 minute break and/or lunch/coffee break during your interviews, especially if you'll be interviewing for more than 3 hours. That said, we are happy to accommodate additional breaks or fewer breaks, depending on your preferences, just let your recruiter or recruiting coordinator know!
- To recent mothers, we are happy to accommodate any breaks needed during your video calls. Please be sure to inform your recruiting partner, so we can make the arrangements that are best for you!

Accommodations:

- If you need any accommodations to make your interview process a success, please don't hesitate to ask.

—

Thank you again for taking the time to interview with 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!