

Back End Developer interview questions

careertips.com.au: common questions and what they're really testing

What to expect

Back end developer interviews usually mix technical depth with judgement calls, since the role sits behind the scenes but still has to hold up under real production conditions. Expect a combination of technical questions on languages, databases and APIs, scenario questions that test how you troubleshoot under pressure, and behavioural questions about working with front-end and DevOps teams.

- **Technical:** Questions on programming languages, database design, API design and cloud deployment, sometimes including a live coding or system design exercise.
- **Scenario:** Judgement-based questions about diagnosing production issues or handling ambiguous requirements with incomplete information.
- **Security-focused:** Questions on authentication, authorization and data protection practices, since back end systems typically handle sensitive data.
- **Behavioural:** Questions about collaborating with front-end developers, DevOps engineers and product teams, and about managing code changes safely in a team.
- **Process:** Questions about your day-to-day workflow, including Git practices, code review and Agile sprint participation.

Most interviews open with a short walk-through of your background and recent projects, move into a technical segment (often a system design or coding exercise, sometimes on a shared screen or whiteboard), then shift to behavioural and scenario questions about how you work with others and handle pressure. There's usually time at the end for you to ask about the tech stack, team structure and deployment practices.

1. Walk me through how you'd design the database schema for a new feature that needs to store and query large volumes of user activity data.

Why they ask

This checks whether you think through normalisation, indexing and scaling trade-offs before writing code, which is central to the role's database design tasks.

How to structure your answer

Walk through your reasoning step by step: identify the data access patterns first, then discuss schema and indexing choices, and finish with any partitioning or scaling considerations for growth.

Example answer

"I'd start by asking how this data will actually be queried, since that shapes the schema more than the data itself does. If it's mostly writes with occasional reporting queries, I'd lean towards a simpler table with good indexing on the fields used for filtering, rather than over-normalising early. I'd also think about whether the volume justifies partitioning by date or user ID down the track, and I'd document the schema and any indexing decisions so the reasoning isn't lost once the feature ships."

2. Tell me about a time you had to debug a production issue under time pressure.

Why they ask

Production troubleshooting is a core part of the role, and this question tests how you stay methodical when something is broken and people are waiting on you.

How to structure your answer

Use STAR: describe the situation, what you were responsible for, the specific steps you took to investigate and fix it, and the outcome.

Example answer

"A service I maintained started returning intermittent errors on one endpoint during a traffic spike. I checked the logs first to narrow down whether it was a code issue or a resource constraint, and found the database connection pool was being exhausted under load. I temporarily increased the pool size to stabilise things, then went back afterwards to fix the underlying query that was holding connections open too long. The incident was resolved within the hour and didn't recur once the query fix went out."

3. How would you approach securing an API that handles sensitive customer data?

Why they ask

Security is listed as a core task for this role, and interviewers want to know you think about authentication, authorization and data protection as part of the build, not as an afterthought.

How to structure your answer

State the security principles you'd apply first, then explain how you'd implement them in this specific context.

Example answer

"I'd start with the basics: enforce HTTPS, use token-based authentication such as OAuth2, and apply role-based authorization so users only access data relevant to them. For sensitive fields I'd look at encryption at rest and make sure logging doesn't capture anything it shouldn't. I'd also want the API tested against common issues like injection and broken access control before it goes anywhere near production data."

4. A front-end developer tells you an API endpoint you built is returning inconsistent data intermittently. How do you investigate?

Why they ask

This tests practical troubleshooting judgement and how you handle ambiguous, hard-to-reproduce bugs, which come up often in integrated systems.

How to structure your answer

Describe your triage process under pressure: how you'd narrow down the cause, who you'd involve, and how you'd communicate status while investigating.

Example answer

"First I'd try to reproduce it myself with the same request parameters, since intermittent issues are often tied to timing, caching or race conditions rather than the code itself. I'd check whether the endpoint hits a cache layer or depends on another service that might be flaky, and look at logs around the reported times. I'd keep the front-end developer updated on what I'd ruled out rather than going quiet, and once I found the cause I'd fix it and add a test that would have caught it earlier."

5. Describe how you've used Git and code review in a team environment to manage changes safely.

Why they ask

This checks whether your day-to-day workflow fits how the team operates, since most back end work happens inside a shared codebase with other developers touching the same services.

How to structure your answer

Describe your usual workflow for branching, commits and review, and how it prevents defects reaching production.

Example answer

"I work off feature branches and keep commits reasonably small so reviewers can actually follow the change. Before opening a pull request I run the existing test suite locally and add tests for anything new. I've found that being specific in the PR description about what changed and why speeds up review a lot, and I try to review other people's code with the same care I'd want on mine rather than rubber-stamping it."

6. What's your experience with containerising and deploying services on a cloud platform such as AWS?

Why they ask

Deployment and monitoring in production environments is one of the role's stated tasks, so interviewers want concrete detail on your hands-on cloud and container experience.

How to structure your answer

Explain your practical experience with the tools, including a specific example of a deployment or incident you handled.

Example answer

"I've containerised services with Docker and run them under Kubernetes for orchestration, mainly on AWS. I'm comfortable setting up the deployment configuration, health checks and basic monitoring so I get alerted if something goes wrong rather than finding out from a user complaint. On one occasion a container kept restarting under load, and tracing it back to a memory limit that was set too tight taught me to test resource limits properly before assuming the application code is at fault."