UnCodemy

SDET Interview Questions PDF

An SDET interview combines coding questions, test design, automation framework design, API and UI testing, CI/CD knowledge and a discussion of quality strategy. Expect to write code, review a test, design a framework on a whiteboard and explain how you would test a feature. This page gives short model answers for practice.

Key takeaways

- SDET roles expect real coding skill plus strong testing thinking.
- Interviews often include a coding task, a test design task and a framework design discussion.
- Know API testing, UI automation, CI integration and how to keep tests reliable.
- Explain trade-offs: what to automate, at which level and why.
- Prepare stories about flaky tests, framework improvements and collaboration with developers.

What is SDET interview questions?

An SDET, or Software Development Engineer in Test, is an engineer who writes code to test software, builds test frameworks and tools, and works with developers to improve quality across the delivery pipeline.

What is the difference between a tester and an SDET?

A manual tester focuses on designing and running tests by hand and reporting defects. An SDET also writes production-quality code for tests and tools, designs frameworks, builds pipelines for tests and may review developers' code for testability. The line between roles differs by company, so read the job description.

What coding questions are common?

Expect problems on strings, arrays, hash maps, sorting, searching and simple recursion, solved in the language the team uses. Typical tasks: reverse a string, check a palindrome, find duplicates, count character frequency, merge two sorted lists, validate balanced brackets. Talk through your approach, write clean code, test it with examples including edge cases such as empty input, and state time and space complexity.

```
Example approach: first non-repeating character
1. Count each character with a map.
2. Scan the string again and return the first with count 1.
Time O(n), space O(k) for k distinct characters.
```

How do I answer test design questions?

Given a feature, clarify requirements, then list tests by category: functional, negative, boundary, security, performance, usability and compatibility. Choose what to automate based on value, stability and cost. Mention data setup, environments and how results will be reported. Show that you think about risk, not only about the number of tests.

What are the questions on frameworks?

How would you design a UI automation framework? Layers for tests, page objects, core utilities, data, configuration and reporting; independent tests; parallel runs; CI integration; clear logs and screenshots. **Which design patterns do you use?** Page Object, factory, builder, singleton or fixtures and strategy, each for a real need. **How do you keep tests maintainable?** Stable locators, small methods, code reviews, removing dead tests and tracking flakiness.

How do I discuss the test pyramid?

Many fast unit tests, a smaller number of API or service tests and a few UI end-to-end tests. Pushing checks lower makes tests faster and less fragile. An SDET helps developers write unit and API tests and keeps UI tests for key user journeys.

What API testing questions should I expect?

Methods and status codes, authentication, request and response validation, schema checks, data setup and cleanup, negative cases, contract testing and how to automate with a client library or a tool, then run in CI. Be ready to describe how to test an endpoint that creates an order, including invalid input and permission checks.

How do I handle flaky tests?

Find the cause: timing, shared data, order dependence, an unstable environment or a real race condition in the app. Fix with condition waits, isolated data, retries only as a short-term measure and mocks for unstable services. Track the flaky rate and fix the worst tests first. Never ignore a flaky test that might point to a real bug.

What CI/CD questions come up?

How tests run on a pull request, how to split smoke and regression suites, how to run tests in parallel and in containers, how to report results, how to treat a failed pipeline and how to manage test environments and data. Mention gates such as "block the merge if critical tests fail".

What about performance and security?

Know the basics of load testing with a tool, reading response time percentiles and bottlenecks, and common security checks such as input validation, authentication and authorisation tests and the OWASP Top Ten categories. Say what you have actually done.

What behavioural questions are asked?

"Tell me about a bug you found that others missed", "describe a framework you improved", "how do you work with developers who disagree with a defect" and "how do you decide what not to automate". Use real stories with the situation, action and result.

How should I prepare?

Practise ten coding problems a week in your language, build a small framework with API and UI tests running in CI, review a famous failure of software quality and think about how tests could have caught it, and rehearse your project stories.

How do I show growth in my answers?

Describe how your work changed over time: from running tests to writing them, from writing tests to designing a framework and from reporting failures to preventing them with earlier checks. Use one real example such as reducing a regression run from hours to minutes by parallelising and removing duplicate checks, stating only facts you can explain.

Frequently asked questions

What is an SDET?

An SDET, or Software Development Engineer in Test, writes code to test software, builds test frameworks and tools and works on quality across the delivery pipeline.

What coding questions are asked in an SDET interview?

Problems on strings, arrays, maps and sorting, such as finding duplicates or counting characters, with attention to edge cases and complexity.

How do I handle flaky tests in an interview answer?

Find the cause, fix it with condition waits, isolated data and mocks, track the flaky rate and never ignore one that may show a real bug.

What is the test pyramid?

It suggests many fast unit tests, fewer API tests and a small number of UI tests, because lower-level tests are faster and less fragile.

Is SDET different from a QA engineer?

An SDET usually needs stronger coding and framework design skills, though titles differ by company.

Sources and further reading

- ISTQB
- Playwright documentation
- Selenium documentation

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550