# Coding Interview — Backend

Coding Interview | Backend

Live coding scorecard for backend: algorithms, data structures, and API implementation.

## Evaluation Criteria

### Problem Analysis & Approach

Evaluates the candidate's ability to understand the problem, identify constraints, and plan an approach before coding.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- What are the inputs and outputs? What constraints should we consider?
- What's the brute force approach, and how can we improve it?
- What's the expected time and space complexity of your approach?

✓ STRONG SIGNAL

Candidate clarifies requirements, discusses multiple approaches with trade-offs, identifies edge cases (empty input, duplicates, overflow), and states the complexity of their chosen approach before coding.

✕ WEAK SIGNAL

Candidate jumps into coding without understanding the problem, cannot articulate their approach, or ignores edge cases entirely.

### Data Structure & Algorithm Selection

Assesses ability to choose appropriate data structures and algorithms for the problem at hand.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- Why did you choose this data structure over alternatives?
- What's the time complexity of this operation, and could we do better?
- How would your approach change if the dataset were 100x larger?

✓ STRONG SIGNAL

Candidate selects optimal data structures with clear justification (hash map for O(1) lookup, heap for top-K, etc.). They understand time-space trade-offs and can adapt when constraints change.

✕ WEAK SIGNAL

Candidate defaults to arrays and nested loops for everything, cannot analyze complexity, or chooses data structures that make the problem harder than necessary.

### Code Implementation Quality

Evaluates the cleanliness, correctness, and organization of the code produced during the session.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- Can you walk me through your code line by line?
- Is there a way to make this function more readable?

- How would another developer understand this code without comments?

✓ STRONG SIGNAL

Candidate writes modular, readable code with descriptive naming. Functions have single responsibilities, error handling is present, and the code works on first or second attempt with minor bugs.

✕ WEAK SIGNAL

Code is a single monolithic function with cryptic names, no error handling, numerous bugs that the candidate cannot debug, or fundamentally incorrect logic.

### Testing & Edge Case Handling

Assesses the candidate's approach to verifying correctness and handling boundary conditions.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- How would you test this solution?
- What edge cases might break your code?
- Can you trace through your code with this input?

✓ STRONG SIGNAL

Candidate proactively identifies edge cases (null/empty input, single element, maximum values), traces through their code manually, and writes or describes test cases without being prompted.

✕ WEAK SIGNAL

Candidate declares the solution 'done' without testing, cannot trace through their own code, or misses obvious edge cases like empty input.

### Communication & Collaboration

Evaluates how well the candidate communicates their thought process and responds to hints or guidance.

Rating: □ 1 □ 2 □ 3 □ 4 □ 5

**Sample Questions:**

- Can you think out loud as you work through this?
- What if I told you there's a way to do this without sorting?
- Where are you stuck, and what options are you considering?

✓ STRONG SIGNAL

Candidate narrates their thinking clearly, takes hints productively and builds on them, asks good clarifying questions, and treats the interview as a collaborative problem-solving session.

✕ WEAK SIGNAL

Candidate codes in silence, ignores or rejects hints, becomes defensive when asked questions, or cannot explain the code they just wrote.

## Red Flags

- Cannot implement a basic algorithm (sorting, searching, string manipulation) in their language of choice
- Memorized solutions to specific problems but cannot adapt when the problem is slightly modified
- Refuses to discuss complexity or cannot explain Big-O notation at a basic level

## Notes & Overall Recommendation

□ Strong Hire □ Hire □ No Hire □ Strong No Hire

Notes: _____