SAMPLE REPORT · Fictional candidate, shown to illustrate the format and depth of a real report

CANDIDATE	TARGET ROLE	END CLIENT	PREPARED FOR
John Doe	Senior C# Developer, FX eTrading (Front Office)	Tier-1 investment bank, HK	[Your firm]
CURRENT POSITION	**DECLARED EXPERIENCE**	**INTERVIEW**	**FORMAT**
Senior Developer, eFX pricing	9 years, 6 in front-office technology	75 min, video	Oral + 20 min live coding

RECOMMENDATION

GO

Submit to the client, after the targeted preparation on page 4

No-go · Hold · **Go** · Strong go

A genuine senior C# engineer with real FX pricing experience. His knowledge of the language, the runtime and the FX workflow holds up under detailed follow-up questions, and he communicates like someone who has sat next to a trading desk.

The one material gap is **low-latency engineering**: solid on everyday concurrency, thin on lock-free techniques and latency measurement. This matters little for pricing and RFQ teams and a lot for a market-making team.

3.2/4
Overall, weighted for the role

Assessed level
Senior
consistent with CV

COMPETENCY PROFILE

Scale 1 to 4, where 3 means "meets the bar for this role". Evidence for every score on pages 2 and 3.

IT fundamentals	**3.5**	Strong and precise, explains from first principles
C# and .NET	**3.5**	Deep on memory, GC and async; lighter on newest APIs
Design and architecture	**3.0**	Sound designs; failure modes needed prompting
Concurrency and low latency	**2.5**	Main gap: no lock-free or measurement practice
SQL and data	**3.0**	Writes correct analytical queries unaided
FX and markets knowledge	**3.0**	Fluent on spot, forwards and FIX; options basic
Logic and problem solving	**3.5**	Structured, states assumptions, checks edge cases
Live coding	**3.0**	Working, tested solution in 14 minutes
Communication	**3.5**	Clear, concise, says "I don't know" when true

WHAT YOU NEED TO KNOW

Strengths

- Runtime knowledge tied to real production incidents, not textbook answers
- Understands the FX quote-to-trade lifecycle end to end
- Honest about limits, which bank interviewers reward

Risks

- Low-latency depth below what the CV wording suggests
- "40% latency reduction" claim not backed by a measurement method
- Thread safety was an afterthought in the coding exercise

The client will probably probe

- How exactly he measured latency (percentiles, tooling)
- False sharing, memory barriers, lock-free queues
- Behaviour of his design when a consumer is slow

Integrity check
Identity and assistance

✓ Identity matched against photo ID and LinkedIn
✓ Entire desktop shared during live coding
✓ Exercise changed mid-way, adapted without delay
✓ Camera on for the full session
✓ Unscripted follow-ups answered consistently
No sign of AI or third-party assistance

John Doe
Senior C# Developer, FX eTrading

✓ answered correctly and in depth △ partial, or correct only after a hint ✕ not known or incorrect

1 IT fundamentals — 3.5 / 4

Stack and heap · Complexity · Hash maps · Processes and threads · TCP and UDP · Virtual memory

- ✓ Explained hash collisions, resizing and why lookups are O(1) on average but O(n) in the worst case, without being asked.
- ✓ Clear on process versus thread, and on why market data is usually sent over UDP multicast while orders go over TCP.
- ✓ Chose the right structure for each scenario given (sorted levels, lookup by ID, FIFO queue) and justified the cost.
- △ Virtual memory and paging: right idea, vague on page faults and their effect on latency.

2 C# and .NET — 3.5 / 4

Value and reference types · Boxing · GC generations, LOH · async / await · IDisposable · LINQ · Span<T>

- ✓ Described Gen 0/1/2 and Large Object Heap fragmentation, then linked it to a real incident: Gen 2 pauses on a pricing service, fixed by pooling buffers.
- ✓ async/await: explained the compiler-generated state machine, *ConfigureAwait(false)*, and the classic *.Result* deadlock on a UI thread.
- ✓ Spotted the hidden boxing and the deferred LINQ execution in the two code snippets shown on screen.
- △ *Span<T>* and *ref struct*: knows what they are for, has not used them in production.

> **Interviewer's note.** Answers stayed accurate three follow-up questions deep. This is the level of a developer who has debugged the runtime, not one who has revised a list.

3 Design and architecture — 3.0 / 4

SOLID · Strategy, Observer, Factory · Dependency injection · MVVM · System design: price distribution

- ✓ Gave a Strategy example from his own code (pluggable spread models per client tier) rather than a textbook one.
- ✓ Explained dependency injection through testability: swapping the live rate source for a replay in tests.
- △ System design ("distribute prices to 500 client sessions"): sensible components, but slow consumers, conflation and backpressure came only after a prompt.
- △ Defaulted to a Singleton for shared configuration; acknowledged the testing cost when challenged.

4 Concurrency and low latency — 2.5 / 4

lock, Interlocked · Deadlocks · Concurrent collections · False sharing · Lock-free queues · GC-free hot path · p99 latency

- ✓ Solid on locks, deadlock causes and prevention, *Interlocked*, and when to use *ConcurrentDictionary*.
- △ Knows object pooling and allocation avoidance on the hot path, applied them once under a senior colleague's lead.
- ✕ Could not explain false sharing or cache-line padding.
- ✕ No hands-on experience with lock-free structures (ring buffer, Disruptor pattern); answers became generic.
- ✕ Could not say how latency was measured on his project: no mention of percentiles, histograms or tooling.

> **How to read this score.** Sufficient for pricing, RFQ and trader-tools teams, where latency is counted in milliseconds. A real gap for a market-making or execution team working in microseconds. Check which one the client is hiring for before you submit.

5 SQL and data — 3.0 / 4

Joins · Indexes · Window functions · Execution plans · Transactions and isolation

- ✓ Wrote "latest trade per currency pair" correctly with *ROW_NUMBER() OVER (PARTITION BY ...)* on the first attempt.
- ✓ Clustered versus non-clustered indexes, and why an index speeds reads but slows writes.
- △ Isolation levels: named all four, unclear on phantom reads and on what snapshot isolation changes.

✓ answered correctly and in depth △ partial, or correct only after a hint ✕ not known or incorrect

6 FX and markets knowledge — 3.0 / 4

Spot, forward, swap | Pips, bid / offer | Value dates | RFQ and streaming | Last look | FIX protocol | Options basics

- ✓ Derived forward points from the interest-rate differential and explained why a forward is not a forecast.
- ✓ Walked through both RFQ and streaming price flows, from liquidity-provider quote to client fill, including tiered spreads and last look.
- ✓ FIX: NewOrderSingle, ExecutionReport, heartbeats, and how a sequence gap is recovered with a resend request.
- △ Spot value date: knew T+2, did not know the T+1 exceptions (USD/CAD for instance).
- △ Options: delta at a basic level only, no knowledge of the volatility surface. Outside the scope of this role.

7 Logic and problem solving — 3.5 / 4

- ✓ **Crossed-book puzzle** (when can the best bid be above the best offer, and what should the system do?): listed stale quotes, different venues and clock skew, then proposed a handling rule.
- ✓ **Estimation** (messages per second for 40 pairs across 8 providers at peak): stated assumptions, reached a defensible order of magnitude, sanity-checked it.
- △ Jumped to a solution on the second puzzle before clarifying the constraints; corrected himself within a minute.

8 Live coding — 20 minutes, his own IDE, entire desktop shared — 3.0 / 4

Exercise. Aggregate a stream of FX quotes from several liquidity providers and expose the best bid and best offer per currency pair.

Change introduced at minute 12. Quotes older than 500 ms must now be ignored.

- ✓ Working solution in 14 minutes, readable naming, no dead code.
- ✓ Absorbed the change in 4 minutes with a timestamp check; no rewrite needed.
- ✓ Wrote two unit tests unprompted, including the stale-quote case.
- △ Not thread-safe. Saw it when asked, proposed a lock, then discussed swapping an immutable snapshot.
- △ Rescans all providers on every read: O(n) where a maintained best level would be O(1).

```
public Quote? BestBid(string pair)
{
    var now = _clock.UtcNow;
    Quote? best = null;
    foreach (var q in _byPair[pair].Values)   // ①
    {
        if (now - q.Time > _maxAge) continue; // ②
        if (best is null || q.Bid > best.Bid)
            best = q;
    }
    return best;
}
```

① Plain *Dictionary* read while another thread writes: the thread-safety gap noted opposite.
② Clean handling of the mid-exercise change. The injected clock made it testable, a good habit.

> **Interviewer's note.** He thinks aloud, types his own code and reacted to the changed requirement immediately. This is the strongest evidence in the session that the experience on the CV is his own.

9 Communication and client-facing behaviour — 3.5 / 4

- ✓ Answers in two or three sentences, then offers to go deeper. Easy to interview.
- ✓ Said "I don't know" three times, each time followed by how he would find out. No attempt to bluff.
- ✓ Explained a technical trade-off in business terms when asked to "say it as you would to a trader".
- △ Undersells his own contribution ("we did") on team projects; needs to state clearly what he built himself.

John Doe
Senior C# Developer, FX eTrading

CV CLAIMS CHECKED DURING THE INTERVIEW

CLAIM ON THE CV	RESULT	WHAT SUPPORTED IT
Led the migration of the pricing GUI to WPF / MVVM	Confirmed	Detailed on the dispatcher, binding throttling and the first release's problems
FIX connectivity to six liquidity providers	Confirmed	Session-level detail, provider quirks described from memory
Reduced tick-to-quote latency by 40%	Partly confirmed	Described the changes made; could not explain how the 40% was measured
Expert in multithreading	Partly confirmed	Competent practitioner; "expert" overstates it (see section 4)

PREPARATION PLAN BEFORE THE CLIENT INTERVIEW

About 6.5 hours of work. Shared with the candidate in the Assess + Prepare package.

	TOPIC	WHAT TO BE ABLE TO DO	EFFORT
1	**Latency measurement**	Explain p50 / p99, why averages mislead, and prepare a factual account of how the 40% improvement was measured, or reword the CV line	1 h
2	**False sharing and cache lines**	Explain the problem with a two-counter example and how padding fixes it	1.5 h
3	**Lock-free basics**	Describe a single-producer single-consumer ring buffer and when it beats a lock; know what compare-and-swap does	2 h
4	**Slow consumers in system design**	Bring conflation and backpressure into the design answer without being prompted	1 h
5	**Span<T> and ArrayPool**	Write a small allocation-free parser to speak from practice	1 h

Advice for positioning the candidate

- Lead with FX pricing and FIX experience; it is verified and directly relevant.
- Present him for pricing, RFQ or trader-tools teams. For a microsecond-level execution team, tell the client about the gap up front.
- Ask him to replace "expert in multithreading" with a description of what he built.

Follow-up included with Assess + Prepare

- 45-minute preparation session with the candidate on topics 1 to 4
- Short re-test on the weak areas, with an updated score sent to you
- Debrief after each client round, and adjustments before the next

HOW THE SCORES ARE SET

1 Below the bar
Basic notions missing or wrong. Would fail a client interview on this topic.

2 Partial
Knows the vocabulary, cannot go past the first follow-up. A gap to close before submission.

3 Meets the bar
Correct and practical answers at the level the role requires, backed by real experience.

4 Exceeds
Depth beyond the role. Could teach the topic or lead on it.

Scores are calibrated against the target role and seniority, not against an abstract standard: a 3 for a senior front-office role is a harder mark than a 3 for a junior one. The overall score weights each area by its importance for the role as described in your job specification.

HOW THE INTERVIEW WAS RUN

Built for this role

Questions were prepared from your job description and the candidate's CV. No fixed question bank, so answers cannot be found online or rehearsed.

Led by a practitioner

The interviewer has worked as a front-office developer in tier-1 investment banks and asks what a hiring manager on a trading floor would ask.

Evidence, not impressions

Every score is tied to what the candidate said or wrote. The session recording and transcript are available to you on request for 90 days.

SCOPE AND CONFIDENTIALITY

This report reflects a single interview and is one input to your decision, not a guarantee of the candidate's performance with the end client. It is prepared for the named recipient and may be forwarded to the end client in full or as the one-page summary; a white-label version under your firm's branding is available. The candidate was informed of the purpose of the interview, of the recording, and of who receives this report.