MCP SERVER | NO CODE | CLOUD HOSTED

CoderPad MCP for AI Agents

Manage technical coding assessments and interview event logs

CoderPad lets you take full command of your technical hiring pipeline directly from any AI client. You can automatically create live coding pads, track candidate keystrokes, and audit entire interview sessions without opening a dashboard. It keeps all assessment data—from the question bank to user roles—in one conversational flow.

F Quality Score 3.6/100

technical-interview | coding-assessment | remote-hiring | collaborative-coding | hiring-workflow | candidate-evaluation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/coderpad

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

CoderPad MCP

8 tools available
Cloud-hosted on Vinkius

Connecting your CoderPad account lets you manage complex technical interviews through natural conversation. Instead of clicking between separate dashboards, you talk to your agent, and it handles the logistics for preparing, running, and reviewing candidate assessments. You can ask your AI client to generate a new coding pad in Python for an immediate interview, or retrieve a full play-by-play log showing exactly when the candidate typed specific code blocks. The platform also lets you audit your organization's question bank and list all current users involved in hiring. This power is available through Vinkius, giving your AI client access to 4,000+ tools across industries, making CoderPad just one part of a massive workflow capability.

Core Capabilities

01 — Create Live Coding Pads

Instantly generates a new collaborative coding environment for an interview session.

02 — View Session Details and Code

Retrieves detailed information about a specific pad, including the candidate's current code content.

03 — Audit Interview Actions

Provides a step-by-step log of everything that happened in an interview session, like keystrokes and executions.

04 — Manage Candidate Questions

Lists all available technical questions stored securely in your organization's question bank.

05 — List All Team Users

Retrieves a list of every user and interviewer within the CoderPad account.

06 — Review Pad Usage History

Accesses records detailing how often pads are used and what quota is consumed across the organization.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/coderpad** — connect your AI agent in three steps.

01 First, subscribe to this MCP on Vinkius.

02 Next, provide your CoderPad API Key (you'll find it in Settings > API).

03 Finally, start managing the entire technical interview process using your preferred AI client.

The bottom line is that you manage assessment workflows entirely through chat commands, without ever having to open the native CoderPad dashboard.

Built For

This MCP is for hiring managers and technical recruiters who are tired of manually juggling dashboards and spreadsheets. If your process requires auditing keystrokes or coordinating multiple interviewers, this tool streamlines that mess.

Technical Recruiter

Uses the MCP to automate pad creation for candidates and verify their progress in real time using natural language prompts.

Hiring Manager

Audits event logs and reviews code submissions immediately, getting a complete picture of the interview without having to open the candidate's specific session.

Software Engineering Lead

Prepares complex interview environments and quickly pulls up relevant questions straight from their chat interface when needed.

What Changes When You Connect

01 Audit every action taken during a session. Instead of guessing what happened, you can use the `get_pad_event_log` tool to see a complete play-by-play of keystrokes and executions.

02 Stop losing track of candidates. You can easily create new pads using `create_new_interview_pad` and keep them tracked right alongside your hiring notes.

03 Never forget a question. The MCP lets you pull up the entire library by listing questions via `list_coderpad_questions`, making sure every candidate is tested against your standard bank.

04 Get instant status updates. You can use `get_pad_session_details` to check a pad's current code contents and its overall status without needing to open any web browser.

05 Simplify team coordination. With `list_coderpad_org_users`, you quickly verify who has access, whether it's an interviewer or an admin, keeping your system secure.

06 Keep track of costs and limits. Review the usage history using `get_coderpad_usage_history` to monitor resource consumption across all technical assessments.

Real-World Applications

The Hiring Manager Needs a Full Audit

A hiring manager needs proof that a candidate truly understood the system design challenge. They ask their agent to retrieve the play-by-play log of all actions in a specific pad, confirming not just what code was submitted, but *how* it was typed and executed using `get_pad_event_log`.

The Recruiter Needs Immediate Setup

A recruiter has an interview scheduled for 2 PM. They ask the agent to create a new Python pad immediately, specifying the role. The agent runs `create_new_interview_pad` and provides the link instantly, skipping manual setup time.

The Engineering Lead Needs Benchmarking

An engineering lead wants to ensure they are asking standardized questions across all roles. They prompt the agent to list available interview questions via `list_coderpad_questions` and compare it against a checklist of required skills.

The Ops Team Needs Compliance Reporting

An ops team member must check who has access to the hiring tools. They ask the agent to list all organization users via `list_coderpad_org_users` and review usage history using `get_coderpad_usage_history` for compliance.

Patterns to Avoid

Using Spreadsheets for Logs

✗ AVOID

A manager tries to track candidate code submissions by manually copying and pasting session data into a Google Sheet, resulting in messy, hard-to-read text blocks.

✓ INSTEAD

Instead, ask your agent to use `get_pad_event_log` or `get_pad_session_details`. This pulls structured, clean data directly into the chat interface for analysis.

Forgetting Question Standardization

✗ AVOID

A team member starts writing unique questions ad-hoc in a meeting, meaning that candidates are assessed differently every time.

✓ INSTEAD

Use `list_coderpad_questions` to pull from your secure, centralized question bank. This ensures consistency and makes auditing easier.

Guessing Candidate Progress

✗ AVOID

A recruiter only checks the main CoderPad dashboard summary, which doesn't show if a candidate is stuck on a specific line of code or hasn't completed the task.

✓ INSTEAD

Ask your agent to run `get_pad_session_details`. This provides granular insight, showing exactly what code was last viewed and the session status.

The Right Fit

Use this MCP if your technical hiring process relies on auditing deep activity. If you need to see *when* a candidate typed something or how many times they hit 'run', CoderPad is essential because it captures the event log. Don't use it if you only need basic communication (like scheduling an interview). For simple task management, an external calendar tool is better. However, if you just want to know *who* was interviewed and nothing else, listing sessions via `list_coderpad_sessions` gives you that high-level overview, but the deep audit capability remains its strongest selling point.

CoderPad for Technical Interviewing: Auditing Candidate Keystrokes

Currently, tracking a technical interview is a mess. You jump between pads, trying to figure out if the candidate struggled with a specific data structure or if they simply forgot something. You rely on vague notes like 'seemed slow' or 'had trouble with X,' which aren't measurable.

With this MCP, you tell your agent to retrieve the play-by-play log of actions for any pad. It gives you an accurate audit trail of every keystroke and execution event. You stop relying on guesswork; you have verifiable data.

CoderPad for Candidate Assessment: Standardizing Question Banks

Manual hiring often means that different interviewers use slightly varied question sets, making it impossible to compare candidates fairly. You waste time trying to standardize the assessment criteria after every single hire.

Now, you can simply ask your agent to list available questions from your organization's bank. This capability ensures every candidate is measured against the same, controlled set of benchmarks, giving you a consistent talent pipeline.

8 CoderPad Tools for AI Agents: Technical Interviewing Assessment

Use these tools to manage the entire lifecycle of technical interviews, from creating new pads to auditing detailed session logs and reviewing question banks.

#	TOOL	DESCRIPTION
01	`create_new_interview_pad`	Generates a brand new live collaborative coding pad for an interview.
02	`get_my_coderpad_profile`	Pulls basic information about the user who authenticated the MCP.
03	`get_pad_event_log`	Retrieves a full, chronological log of every action taken within a specific interview pad.
04	`get_pad_session_details`	Gets comprehensive details for a given coding pad, including the candidate's current code contents.
05	`list_coderpad_org_users`	Retrieves a list of all users and interviewers associated with the organization account.
06	`list_coderpad_sessions`	Lists every technical interview pad session that has occurred in the organization.
07	`list_coderpad_questions`	Shows all available practice and assessment questions from the centralized question bank.
08	`get_coderpad_usage_history`	Retrieves a history report detailing pad usage counts and quota consumption.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U List all my CoderPad sessions from this week.

Retrieving your recent sessions... Found 5 pads:

***Backend Dev - John Doe**:* Live*. Status: In Progress. Last Activity: 10 minutes ago.
***Frontend Role - Sarah Smith**:* Completed*. Status: Submitted. Code Review Available.
***Data Scientist - Alex Kim**:* Completed*. Status: Abandoned.

Do you want the current code contents for John Doe's session, or do you need to review the event log for Sarah Smith?

U Create a new Python pad for 'Junior Engineer Interview'.

✅ Python coding pad created successfully.

Session Details:
• **Title**: Junior Engineer - Python Challenge
• **Language**: Python 3
• **Link**: [https://coderpad.io/session-xyz]

I can pre-populate the environment with a standard question if you'd like, or I can just leave it clean for the candidate to start fresh.

U Show me all available database questions.

Here are 5 of the most recent and relevant questions from your Question Bank:

1. **SQL: Top Customers**: Write a query to find the top 5 customers by revenue.
2. **System Design: URL Shortener**: Outline the necessary tables and relationships for a short URL service.
3. **NoSQL: Product Catalog**: How would you index product search data in MongoDB?
4. **Advanced SQL**: Write a query using window functions to calculate rolling averages.
5. **Indexing Strategy**: Explain why proper indexing is crucial for database performance.

Frequently Asked Questions

01 **How do I audit what candidates were doing during the CoderPad interview?**

You can retrieve a full, detailed log of every action taken in a session. This means seeing specific keystrokes and execution attempts, giving you deep insight into their thought process beyond just the final code.

02 **Does CoderPad help me keep my interview questions consistent?**

Yes, it keeps all your available questions in a central bank. You can list these standard questions to ensure every candidate is assessed using the same set of benchmarks across different roles.

03 **How do I manage user access for my interview team?**

The MCP allows you to list all users and designated interviewers within your organization's CoderPad account, helping you keep track of who has permissions to run assessments.

04 **Can CoderPad help me set up a new coding assessment quickly?**

Absolutely. You can ask the agent to create a brand new live pad in any language (like Python or JavaScript) instantly, generating the link and readying the environment for the candidate right from your chat.

05 **Is CoderPad suitable for tracking multiple assessment types?**

Yes. It handles more than just code submissions; you can also track usage history to monitor how many assessments are run and manage quota consumption across teams.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"coderpad": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

CoderPad is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by CoderPad. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	CoderPad MCP
Server ID	`019d7576-9a0c-7137-95f2-3615b811d2d2`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/coderpad.