MCP SERVER | NO CODE | CLOUD HOSTED

Xray Test Management MCP

Get QA insights through plain conversation.

Xray (Test Management) MCP connects your AI agent directly to Xray, Jira's leading quality assurance platform. Use natural conversation to manage complex testing workflows. You can list all test cases, check execution results for specific runs, monitor entire test plans, and audit historical data across projects—all without navigating Jira menus.

A+ Quality Score 100/100

test-management | qa-automation | jira-integration | test-execution | bug-tracking | software-quality

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/xray-test-management

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.

Xray (Test Management) MCP

9 tools available
Cloud-hosted on Vinkius

Managing QA is usually a nightmare of dashboards, filters, and deep linking. This MCP changes that by letting your AI agent talk to Xray directly. Instead of spending time clicking through complex issue lists just to find out why a test failed, you simply ask your client what happened. Your agent acts as an experienced QA analyst, pulling granular data on everything from specific failing steps to overall release readiness. You can check the status of large test plans or audit how one test case performed over several cycles. Because this MCP is hosted and managed by Vinkius, you connect once through any compatible client—Claude, Cursor, Windsurf, etc.—and get immediate access to all these powerful testing tools. It's about making your AI agent do the heavy lifting so you can focus on fixing the bugs.

Core Capabilities

01 — Discover and Inspect Test Cases

List all available test cases in a project and pull out specific unique keys for deep inspection.

02 — Monitor Execution Results

Track current QA outcomes by listing and checking records from completed test runs to gauge release health.

03 — Analyze Failed Test Steps

Get detailed results for a specific execution, pinpointing exactly which steps passed, failed, or hit an error.

04 — Review Test Strategies and Plans

Browse high-level test plans to understand the scope of testing and track overall progress against goals.

05 — Organize Tests into Groups

See how individual test cases are grouped together in functional or regression test sets.

06 — Audit Test History

Retrieve the full run history for any single test case across multiple cycles to identify unstable or flaky tests.

07 — Verify Project Configuration

Check environment settings and status mappings to ensure your AI agent is aligned with the project's workflow.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/xray-test-management** — connect your AI agent in three steps.

01 Subscribe to this MCP using your Xray Client ID and Secret.

02 Connect your preferred client (Claude, Cursor, etc.) through Vinkius.

03 Ask your agent a question in plain English; it uses the available tools to fetch and summarize complex test data.

The bottom line is you get instant, conversational access to deep QA metrics without manual dashboard navigation.

Built For

This MCP is essential for Quality Assurance Engineers, Test Managers, and Product Owners who are tired of copy-pasting data from Jira dashboards. If checking release readiness or debugging a flaky test requires clicking through five different screens, you need this.

QA Engineer

They use the MCP to check specific execution results and pull historical data on individual tests to help pinpoint failure causes.

Test Manager

They rely on it to quickly surface progress across entire test plans and verify how test sets are organized without manual navigation.

Product Owner

They use the MCP to check the latest execution status before a release is greenlit, ensuring quality standards were actually met.

What Changes When You Connect

01 Audit historical data for flaky tests. Instead of manually tracking a test's performance across multiple Jira runs, you use the tool to get the complete run history and spot patterns instantly.

02 Know exactly what failed. If an execution fails, don't just see 'Failed.' Use the function that retrieves granular results to pinpoint if the problem was in step three or the payment submission logic.

03 Plan reviews are faster than ever. Get test plan progress by listing all available plans and seeing the current completion percentage without navigating multiple boards.

04 Understand your scope quickly. You can list all test cases, giving you a quick overview of what's being tested, and then use related tools to see how they're grouped in functional sets.

05 Check project alignment easily. Use the function that verifies environment settings to confirm your agent is reading status codes correctly for the current build target.

Real-World Applications

Investigating a Critical Failure

A QA Engineer finds an issue during staging but can't tell if it's intermittent. Instead of spending hours comparing logs, they ask their agent to list individual test runs for that specific test case key and check the run history to see if this failure pattern is new or persistent.

Release Readiness Check

A Product Owner needs sign-off for a major version release. They ask their agent to list all active test plans and summarize which ones are 90% complete, getting an immediate status report without opening the main dashboard.

Debugging Development Issues

A developer is debugging a failure reported by QA. Instead of asking QA for screenshots, they ask their agent to get execution details for that specific run, which immediately shows them the exact step and error code (e.g., 500) needed to fix it.

Validating Test Coverage

A Test Manager wants to ensure a new feature wasn't missed. They ask their agent to list test sets related to the 'Billing Module,' which shows them all the functional groups they need to review and update.

Patterns to Avoid

Treating it like a search bar

✗ AVOID

Typing, 'I want to know about the mobile login test.' This vague query forces the agent to guess which specific tool or dataset you mean.

✓ INSTEAD

Instead, ask: 'List all test cases in the Mobile-App project and give me their keys.' Use the `list_xray_tests` function for a precise list of available assets.

Asking for generalized reports

✗ AVOID

Saying, 'Give me an overall report on last week's testing.' This is too vague and requires manual filtering by the agent.

✓ INSTEAD

Be specific: 'List test executions from the past seven days that had a failure count greater than zero.' Use `list_test_executions` to narrow down the search.

Confusing plans with sets

✗ AVOID

Asking, 'Show me the plan for regression testing.' This might pull general documentation instead of the current execution status.

✓ INSTEAD

Use `list_test_sets` first to see how tests are grouped, then use a specific tool like `get_test_plan_details` with the correct key.

The Right Fit

You should use this MCP if your primary job involves auditing, tracking, or reporting on quality assurance outcomes from Jira's Xray platform. It excels when you need to move beyond a simple 'pass/fail' status and dive into the mechanics of *why* something failed—like checking specific steps within an execution using `get_execution_details` or understanding test history with `list_individual_test_runs`. Don't use this if your goal is simply creating new test cases, writing documentation, or managing user permissions. For those tasks, you need a dedicated Jira administrative tool, not a QA reporting MCP.

The Manual Pain of Quality Assurance Reporting

Right now, checking release status means logging into Xray and navigating through dozens of dashboards. You have to manually filter by date range, then select the specific test plan, check if all required test sets are complete, and finally copy-paste summary data for stakeholders. It's a painful sequence of clicks.

With this MCP, you just talk to your agent. Instead of clicking through five screens to get that status, you ask: 'Show me the progress for v2.4 Release.' The answer comes back immediately in plain language, giving you instant visibility into test plan completion and failure metrics.

Xray Test Management MCP Delivers Deep Insights

The tedious steps of manually comparing execution results to determine if a bug is flaky or critical disappear. You can ask the agent to list test executions and then use specific tools like `get_test_details` to pull up all necessary context for review.

Your AI client now acts as an embedded QA specialist, giving you immediate answers about your codebase's health that were previously locked away behind complex Jira menus. It’s pure, instant intelligence.

Xray (Test Management) with 9 Tools

These tools let your agent pull specific data from Xray, giving you the power to track test progress, view detailed execution results, and manage project configurations conversationally.

#	TOOL	DESCRIPTION
01	`get_xray_settings`	Fetches the current project configuration and field mappings for Xray.
02	`get_execution_details`	Retrieves detailed results for a specific instance of test execution.
03	`get_test_plan_details`	Gets comprehensive information about a defined test plan.
04	`get_individual_test_runs`	Pulls individual run instances for a specific test case key.
05	`get_test_details`	Retrieves all available details for any specified Xray test case.
06	`list_test_executions`	Lists every recorded instance of a test execution that has taken place.
07	`list_test_plans`	Shows a list of all active and configured test plans within the project.
08	`list_test_sets`	Lists functional or regression groups, showing how tests are organized together.
09	`list_xray_tests`	Provides a complete list of every test case defined in the Xray project.

See It in Action

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

U List all test cases in the 'Mobile-App' project.

I found 15 test cases in project 'Mobile-App'. The most recent are: 1. 'Login Validation' (Key: TST-101), 2. 'Profile Update' (Key: TST-102), and 3. 'Push Notifications' (Key: TST-103). Which one would you like to inspect?

U What were the results for execution 'EXEC-555'?

Results for EXEC-555: Total Tests: 10. Passed: 8, Failed: 2 (TST-42, TST-45). Failed steps in TST-42: 'Submit payment' returned a 500 error. Would you like me to pull the logs for these failures?

U Show me the test plan progress for 'v2.4 Release'.

Plan 'v2.4 Release' (Key: PLAN-12) is 75% complete. 45/60 tests have been executed. Current status: 40 Passed, 5 Failed, 15 Pending. The regression test set is still pending execution.

Frequently Asked Questions

01 **How do I check if a test case failed repeatedly using Xray Test Management MCP?**

You use the tool to list individual test runs for that specific test case. This lets you retrieve the complete run history, which helps identify flaky tests or recurring failures across multiple cycles.

02 **Can I see a summary of all my available tests using Xray Test Management MCP?**

Yes, simply use the tool to list all test cases in your project. This gives you an immediate overview and unique keys for any deep inspection required.

03 **What is the difference between a Test Plan and a Test Set using Xray Test Management MCP?**

The MCP allows you to list both, but they serve different purposes. You can use `list_test_plans` for high-level strategy and `list_test_sets` to see how tests are functionally grouped together.

04 **How do I find out what failed during a specific test run?**

You retrieve the granular results by using the tool that gets execution details. This pinpoints exactly which steps, not just the overall test, returned an error code or passed.

05 **Does Xray Test Management MCP work with any AI client?**

Yes, because this is an MCP hosted on Vinkius, you connect once from your preferred compatible client (Claude, Cursor, Windsurf) and gain access to all the tools.

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 `"xray-test-management": { "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

Xray (Test Management) 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 Xray (Test Management). 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	Xray (Test Management) MCP
Server ID	`019d7625-b3c4-73c4-a282-097d7e15cdee`
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/xray-test-management.