MCP SERVER | NO CODE | CLOUD HOSTED

UtilityAPI MCP

Analyze Energy Usage & Billing from 100+ Providers

UtilityAPI connects your AI client to billing and usage data from over 100 US utilities—including PG&E, Southern California Edison, and National Grid. It gives your agent access to complete utility billing history, granular 15-minute energy consumption readings, and detailed meter information across multiple providers through one single API.

A+ Quality Score 100/100

billing-data | energy-usage | meter-data | green-button | utility-api | data-aggregation

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

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.

UtilityAPI MCP

12 tools available
Cloud-hosted on Vinkius

This MCP lets you pull comprehensive data from almost any electric or gas utility company in the US, bypassing the headache of dealing with dozens of separate systems. Need to analyze a customer's spending habits? You can retrieve full billing records showing costs and usage amounts across different periods. Want to size a solar array precisely? Your agent pulls granular, 15-minute interval data, identifying exactly when energy consumption peaks. The system also keeps track of which customers have authorized data sharing, ensuring the information you access is legitimate and up-to-date. Instead of building brittle connections for every single provider, you connect once through Vinkius and get instant access to this entire catalog of utility data. This means your AI client can run deep analyses—from calculating annual energy spend to finding peak usage times—all with a single connection.

Core Capabilities

01 — Retrieve Billing History

Your agent pulls complete records showing costs, total kWh or therms used, and the billing date range for any authorized meter.

02 — Analyze Granular Usage Patterns

You get 15-minute or hourly consumption readings from smart meters, allowing you to pinpoint exactly when energy demand peaks occur.

03 — Manage Customer Data Access

Your agent creates and manages the secure authorization forms required for customers to grant data access to their utility records.

04 — Get Meter Status Information

You can list all authorized meters, checking key details like service address, fuel type (gas/electric), and current collection status.

05 — Combine Billing and Usage Data

The system provides a single call that merges both the financial billing records and the detailed time-series usage data for one meter.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and enter your UtilityAPI API Token.

02 Your AI client uses a utility code (found via list_utilities) to generate an authorization form for the customer.

03 Once authorized, you use the meter UID with get_meter_data or get_bills to pull all combined usage and billing records.

The bottom line is your agent gets reliable access to complex energy data from hundreds of utilities using a straightforward workflow.

Built For

This MCP is built for professionals who deal with property assets, energy efficiency, or lending decisions. It helps eliminate the painful process of manually requesting and compiling usage reports from disparate utility websites.

Energy Consultant

Analyzing customer billing history and granular consumption data to write detailed proposals for home or commercial energy efficiency upgrades.

Property Manager

Monitoring utility costs across a portfolio of buildings, identifying high-cost tenants, and flagging unusual anomalies in consumption patterns.

Solar Installer/Engineer

Calculating the required size of solar systems by retrieving historical electricity usage data to accurately forecast ROI for client proposals.

FinTech Lender

Verifying a loan applicant's energy spending and utility payment history as part of an efficiency or property qualification check.

What Changes When You Connect

01 Stop stitching together data. The get_meter_data tool merges bill summaries and interval readings into one call, giving you a complete view of the asset's energy life cycle.

02 Drill down deep on consumption patterns using get_intervals. This access to 15-minute readings lets your agent identify peak demand periods, which is key for optimizing solar sizing or managing facility load.

03 Handle compliance easily. Use create_auth_form and list_utilities to build secure authorization workflows that meet strict data sharing requirements before pulling any records.

04 Speed up analysis with get_bills. Instead of manually collecting invoices month by month, you pull structured billing history instantly, allowing for immediate cost-over-time comparisons.

05 Build robust monitoring. The get_events tool tracks the entire data pipeline—from authorization creation to successful data collection—so you always know if your data feed is ready.

Real-World Applications

Determining Peak AC Usage for a Commercial Build

A property manager needs to know the worst-case load on a building's electrical grid. They ask their agent to use get_intervals, specifying 15-minute readings over a full year. This reveals that peak usage reliably occurs between 3 PM and 6 PM due to HVAC cycling, allowing them to upgrade circuit breakers before an outage happens.

Assessing Loan Eligibility Based on Energy Spending

A lender needs to verify a borrower's stable utility payment history. They use get_bills and list_authorizations to pull 3 years of billing records, confirming consistent payments and usage patterns that meet loan underwriting criteria.

Sizing Solar Systems for Complex Residential Properties

A solar installer needs more than just the last bill. They run get_meter_data to see both the total cost *and* the time-series usage data, identifying periods of peak consumption (e.g., early morning heating spikes) that must be covered by the system.

Auditing Utility Billing for Discrepancies

An energy consultant suspects a utility is overcharging. They use get_intervals to compare recorded usage against the bill dates, finding a mismatch between the expected hourly consumption and the billed kWh amount.

Patterns to Avoid

Treating all data as equal.

✗ AVOID

A user only calls get_bills and assumes they have enough detail to analyze usage patterns. They miss crucial information about *when* the energy was used, leading to inaccurate recommendations.

✓ INSTEAD

Always supplement billing summaries by using get_intervals or get_meter_data. This gives you the time-series data needed to understand peak load times, not just total cost.

Forgetting authorization status.

✗ AVOID

An agent tries to run a usage query on a meter ID that hasn't been authorized recently or has expired credentials. The call fails with generic permission errors and zero data.

✓ INSTEAD

Before running any core functions, check the status using list_authorizations first. If needed, use create_auth_form to guide the customer through updating their access.

Overlooking historical gaps.

✗ AVOID

A consultant only requests current data for a new client and misses potential usage spikes from previous years. The analysis is incomplete because it lacks long-term context.

✓ INSTEAD

After getting customer authorization, run activate_historical_collection immediately. This ensures that when you later call get_meter_data, the historical period is included in your results.

The Right Fit

Use this MCP if your project requires deep, quantitative access to utility-grade data—specifically billing amounts, kWh/therms usage, and time-series consumption patterns. This is non-negotiable for energy modeling, property portfolio analysis, or complex financial verification. Don't use it if you only need a simple 'yes/no' answer about electricity service; just check the list_meters endpoint.

However, don't rely solely on get_bills. If your goal is to find peak usage times (e.g., when the air conditioning runs hardest), you MUST use get_intervals or get_meter_data because billing data only gives totals, not timing.

Tracking Energy Costs and Usage Is a Data Nightmare

Today, analyzing energy costs means jumping through hoops. You start on the utility's website, find your meter number, log in with separate credentials, and manually download PDF bills for the last 12 months. Then you copy-paste consumption numbers into a spreadsheet, trying to align dates, formats, and different unit types (kWh vs therms) before you even begin the analysis.

With this MCP, your agent handles all that complexity automatically. You simply specify the meter ID or range of meters, and the tool retrieves structured data containing both the total cost *and* the granular usage intervals in a clean format ready for immediate calculation.

UtilityAPI Gives You Combined Billing and Usage Data

The biggest manual step that disappears is having to run two separate queries—one for the bill total, and another for the raw usage data. Previously, you needed a billing summary *and* an interval pull just to get a complete picture of one customer's consumption.

Now, calling get_meter_data delivers both financial (billing) and technical (usage patterns) records in a single output, letting your agent build a full energy profile without needing multiple API calls or data merging.

UtilityAPI with 12 Tools

These twelve tools allow your agent to manage authorizations, pull billing records, analyze granular energy usage, and list all associated meter details.

#	TOOL	DESCRIPTION
01	activate_historical_collection	Starts the process of backfilling historical data for specific meters, ensuring your analysis gets more than just current usage.
02	create_auth_form	Generates a unique link for customers to start the secure process of sharing their utility billing information with you.
03	get_billing_summaries	Checks your account's API usage costs and provides general billing period summaries for this MCP.
04	get_bills	Retrieves the official monthly utility bills, including total cost and overall energy consumption amounts (kWh/therms), for an authorized meter.
05	get_events	Monitors system status updates, letting you know when a customer authorization is created or if data collection fails.
06	get_form_templates	Lists the available templates to understand what types of information can be requested during an authorization process.
07	get_intervals	Pulls highly detailed time-series usage data (15-minute or hourly) showing energy flow and cost for thorough demand modeling.
08	get_meter_data	Combines the full bill history with granular interval readings in one call, giving a complete picture of a meter's usage.
09	list_authorizations	Lists all customers who have granted access to their data and provides tracking codes for those records.
10	list_meters	Retrieves a comprehensive list of every authorized service connection, detailing the address and type of utility meter.
11	list_utilities	Provides a directory of all 100+ supported utility companies, including their unique codes for form setup.
12	test_form_submission	Simulates submitting an authorization form using a specific ID to generate test data and verification codes before live use.

See It in Action

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

U List all supported utility companies.

Found 100+ supported utilities including: PG&E (Pacific Gas & Electric), Southern California Edison (SCE), Con Edison, National Grid, SDG&E, LADWP, CenterPoint Energy, Direct Energy, and many more. Each has a unique utility code used for creating authorization forms. Which utility would you like to focus on?

U Get the billing history for meter 44445555.

Retrieved 12 months of billing history for meter 44445555 (PG&E - Residential Electric). Notable bills: Dec 2024: $142.30 (890 kWh), Jan 2025: $156.80 (980 kWh), Feb 2025: $134.50 (845 kWh). Average monthly cost: $144.20. Peak usage in January during winter heating. Total annual spend: $1,730.40.

U Show me the usage intervals for my authorized PG&E meters.

Found interval data for 2 authorized PG&E meters. Meter 44445555 (Electric, Residential): 3,650 interval readings over 30 days, averaging 1.2 kWh per 15-min interval. Peak usage between 2-6 PM (AC usage). Meter 44445556 (Gas, Residential): 720 readings, averaging 0.08 therms per hour. Highest gas usage during morning hours (6-9 AM) for water heating.

Frequently Asked Questions

01 **How does UtilityAPI handle different utility companies?**

UtilityAPI connects to over 100 US utilities through one unified system. You use list_utilities to find the specific code needed for any provider you are working with.

02 **Can I get usage data older than my current bill?**

Yes, but it requires a few steps. Use create_auth_form, then activate_historical_collection to request backfilling of historical data for the meter.

03 **What is the difference between get_bills and get_intervals?**

get_bills gives you the final monthly total cost and usage amount. get_intervals provides the high-resolution, 15-minute or hourly readings needed for detailed time-series analysis.

04 **How do I check if a meter is authorized?**

You can use list_meters to see all connected meters and check their status. For specific customer authorization details, run list_authorizations.

05 **Does UtilityAPI work for commercial buildings only?**

No, it covers both residential and commercial customers. When creating an auth form, you can specify the scenario type to match the property's use.

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 `"utilityapi": { "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

UtilityAPI 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 UtilityAPI. 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	UtilityAPI MCP
Server ID	`019d761a-69aa-7294-a8a4-a45d8576dc3b`
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/utilityapi.