AgentSQL

智盾

MCP 接入指南

v4.6.0

agentsql.cn

v4.6.0 使用文档

目录

v3.9.0 Oracle 与 SQL Server 受控读取

v3.8.0 原生数据库工具

选择传输方式

- Streamable HTTP：已运行服务时首选
- 可选的人员身份委托（v1.0.0 起，v1.1.0 扩展）
- 2026-07-28 无状态 HTTP（v0.7.0 起）
- v1.2.0 可选的 MCP OAuth 入口
- stdio：单机桌面 Agent

MCP 工具

调用时序

- 只读查询
- 写操作：自动转审批
- 写操作：主动送审

正确处理返回值

Claude Desktop 接入

- Windows Claude Desktop：连接已有 AgentSQL 服务

MCP Inspector 本地验收（Streamable HTTP）

Dify 自托管接入（Streamable HTTP）

Cursor 接入（stdio）

豆包及其他远程 MCP 客户端（Streamable HTTP）

从 MCP Registry / Glama 查找

客户端接入安全提醒

自研客户端

- initialize 裸 HTTP 示例
- Go SDK

故障排查

- 401 unauthorized
- stdio 没有控制台 UI
- 只能在服务器本机访问

MCP 握手失败

堆叠多语句被拒绝

写操作提示缺少 reason

stdio 连不上控制面或无法解密数据源

SQLBot

AgentSQL MCP 接入指南

语言 / Language: 简体中文 | English

> 当前正式版为 v4.6.0。先看三种使用方式选择独立 MCP 网关或与自有问数联合使用，再从 v4.6.0 Release核对正式资产、签名与验收边界。本文其余版本化章节保留对应历史能力的具体条件。

本文说明如何把 MCP 客户端接到 AgentSQL。身份、授权、规则、审批与能力边界见 使用手册，首次部署见 快速上手；把决策事件外发到 Webhook、告警群或 Syslog 见 通知外发指南。

当前版本的 `list_datasources` 为当前 Agent 可见的数据源返回 `support` 声明：层级、可用工具、限定范围及证据文档。它只描述已验证的入口，不代替逐次授权。使用仓库提供的演示数据播种文件建立隔离环境后，可运行 `scripts/mcp-acceptance.py` 检查工具发现、允许、R010 拒绝、脱敏和两个持久审计 ID；API Key 仅从 `AGENTSQL_MCP_ACCEPTANCE_KEY` 环境变量读取。示例：

```
AGENTSQL_MCP_ACCEPTANCE_KEY="$DEMO_AGENT_KEY" python3 scripts/mcp-acceptance.py \
  --mcp-url http://127.0.0.1:7780/mcp --datasource-id ds-demo-pg \
  --allow-sql 'SELECT c.id, c.full_name, c.phone, c.email, c.region, o.status FROM
public.demo_b2_customers c JOIN public.demo_b2_orders o ON o.customer_id=c.id WHERE o.id=1' \
  --deny-sql 'SELECT * FROM public.demo_b2_customers WHERE id = 1' --deny-rule R010
```

该示例依赖 `examples/docker/demo-seed.yaml` 中的列级授权剧本；其他环境需换成已授权的允许查询及可被 R010 拒绝的查询。不要把真实客户数据送往公网 Demo。脚本仅输出判定、审计 ID 和脱敏单元格数。

v3.9.0 Oracle 与 SQL Server 受控读取

- 对指定 Oracle 26ai Free 选择 `verify-full`、填写绝对路径 CA 文件，并使用与证书名称匹配的连接主机名。网关每次新建连接均验证证书。Oracle 钱包、mTLS 与 RAC 不在本次验收范围。
- 仅在专用非系统只读账号与已验证的 `PLAN_TABLE` 就绪后，在 `read_control_json` 启用 `{"oracle":{"explain":true}}`。获准的窄 SELECT 随后进行有界计划预检；无法取得计划或计划无效时失败关闭。既有 Oracle 数据源默认仍不启用计划预检。
- Oracle 与 SQL Server 都使用 MCP `query`。SQL Server 保持严格证书校验和限定 SELECT/SHOWPLAN 权限，轮换及恢复后分别核对允许、拒绝、脱敏和持久审计 ID。公网 Demo 仍为 PostgreSQL/MySQL 合成数据。见发布说明、升级指南和验收记录。

v3.8.0 原生数据库工具

- `find_documents` 接收 `datasource_id`、`collection`、显式顶层 `fields`、字符串等值 `filter` 和 1–100 的 `limit`。副本集模式要求过滤字段组合命中管理员批准的索引；禁止聚合、命令和写入。
- `read_hash_fields` 接收 `datasource_id`、精确 `key` 与 1–16 个显式 `fields`。每个 Hash 字段需要独立授权，返回最多 64 KiB，按字段来源脱敏并在持久终态审计后交付；拓扑证据见v3.7.0 验收。
- `query_analytics` 在指定 ClickHouse 26.8.31.2 单节点上可通过管理员 `read_control_json` 收紧查询扫描预算。只读数据库角色必须允许受上限约束的查询级设置；未配置时沿用原预算。
- `search_vectors` 在指定 Milvus 2.6.25 单机上可通过管理员 `read_control_json` 开启权威租户范围；可选 `filter` 仅接收至多四个获准静态字符串字段的等值条件。租户值来自当前 Agent，不由客户端传入。公网 Demo 仍只展示 PostgreSQL/MySQL 合成场景，配置见v3.8.0 设计与验收。

选择传输方式

Streamable HTTP：已运行服务时首选

安装器、systemd 或 Compose 已经运行 AgentSQL 时，直接连接：

- 端点：`http://127.0.0.1:7780/mcp`
- 身份头：`Authorization: Bearer <Agent API Key>`
- MCP 协议版本：`2025-06-18` 或 `2025-11-25`（两者均为旧版握手协议；新版本使用下述独立入口）
- `Accept`：`application/json, text/event-stream`
- 模式：默认 stateful（initialize 返回 `Mcp-Session-Id`）；可显式回退 stateless
- 断线重放：stateful 旧协议默认持久化 SSE 事件；以 `GET /mcp` + `Last-Event-ID` 重连。若事件已因 TTL/容量回收，返回 HTTP 400，不返回部分事件
- 单请求体上限：4 MiB

Agent API Key 不是控制台管理员 token。Agent 被禁用、过期、删除或轮换 Key 后，旧 Key 立即失效。

可选的人员身份委托（v1.0.0 起，v1.1.0 扩展）

管理员可在配置中启用 `mcp.delegation.require_human_token: true`，同时保持 `server.console_enabled: true`。启用后，两个 HTTP MCP 入口的每次请求都要分别带 Agent Key 和人员委托令牌：

```
Authorization: Bearer <Agent API Key>
X-AgentSQL-User-Authorization: Bearer <人员委托令牌>
```

具有 `query.execute` 权限的人员先通过控制台登录（本地账号、LDAP 或 OIDC），再在「任务授权 → MCP 用户身份委托」按 Agent ID 签发；也可用控制台登录令牌调用：

```
curl -fsS -X POST http://127.0.0.1:7780/api/v1/auth/delegation-token \
  -H "Authorization: Bearer ${AGENTSQL_ADMIN_TOKEN}" \
  -H 'Content-Type: application/json' \
  --data '{"agent_id":"<Agent ID>"}'
```

响应里的 `data.token` 只适用于指定 Agent，最长 30 分钟，不能访问控制台 API；普通控制台登录令牌不能直接填入 MCP 第二个请求头。原登录会话退出或失效、用户停用、`query.execute` 被撤销时，委托令牌立即失效。有状态 `/mcp` 会话绑定人员身份，换人需重新 `initialize`。任务级 Key 还要求令牌所属用户与授权签发者一致。令牌签发需要管理审计可写，否则失败关闭。

自 v1.1.0 起，审批申请与执行、B5 跨请求会话也绑定发起人员、Agent 与租户；跨人复用或原人员权限失效时拒绝。请先确认 MCP 客户端可安全配置两个 HTTP 身份头，再启用此开关。公开 Live Demo 不使用这一模式。

进程重启不会恢复 go-sdk 的内存 transport session。携带旧 `Mcp-Session-Id` 的请求会收到 HTTP 404 与 `Mcp-Session-Expired: 1`；客户端应丢弃旧 ID 并重新 `initialize`。新 session 不会读取旧 session 的事件。

2026-07-28 无状态 HTTP（v0.7.0 起）

新协议使用 `http://127.0.0.1:7780/mcp/2026-07-28`。该路径只接受 `2026-07-28`，不会协商为旧版；旧 `/mcp` 的行为和默认配置不变。新路径每个 POST 都重新验证 Agent API Key、由该 Key 确定租户，并与旧路径共用同一 Agent 的 QPS 限额。请求体仍限制为 4 MiB，响应继续遵守网关上限。

客户端每次请求都要发送 `Authorization: Bearer <Agent API Key>`、`MCP-Protocol-Version: 2026-07-28`、`Mcp-Method`、`Content-Type: application/json` 与 `Accept: application/json, text/event-stream`；调用工具时还要发送与 `params.name` 一致的 `Mcp-Name`。`params._meta` 必须包含同版 `io.modelcontextprotocol/protocolVersion` 和 `io.modelcontextprotocol/clientCapabilities`。客户端可先调用 `server/discover` 查询能力，也可直接调用工具；无需 `initialize` 或 `notifications/initialized`。该路径没有 MCP 传输会话、独立 GET 事件流、DELETE 关闭或 `Last-Event-ID` 重放；收到旧 `Mcp-Session-Id` 或 `Last-Event-ID` 请求头时忽略它们。连接断开后由客户端决定是否使用新的请求 ID 重试。详见 MCP 2026-07-28 HTTP 规范 和 Go SDK 协议指南。

新路径对普通 Agent Key 注册下面列出的 12 个基础工具，**暂不提供 B5 跨请求逻辑会话和计划事务工具。**任务级临时只读 Key 只注册 `query`，并且查询仍须通过临时授权和原 Agent 权限。B5 的业务会话与 MCP 传输会话是不同概念；需要 B5 时继续使用旧 `/mcp` 或 stdio。`mcp.http.stateful` 只控制旧路径，不改变新路径的无状态语义。客户端不得把管理员 token 或自报租户头当作 Agent 身份。

v1.6.0 起，这个无状态入口会把 HTTP 客户端断连传给执行中的只读查询；网关在交付结果前再次检查超时或取消状态。旧有状态 `/mcp` 继续按服务端语句超时终止请求，以保留 B5 会话回滚约束。若客户端断连后重试，应使用新的请求 ID，并按审计记录确认前一请求的终态；不要把断连视为查询已经成功。

```
curl -fsS http://127.0.0.1:7780/mcp/2026-07-28 \
  -H "Authorization: Bearer ${AGENTSQL_API_KEY}" \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json, text/event-stream' \
  -H 'MCP-Protocol-Version: 2026-07-28' \
  -H 'Mcp-Method: server/discover' \
  --data '{"jsonrpc":"2.0","id":1,"method":"server/discover","params":{"_meta":
{"io.modelcontextprotocol/protocolVersion":"2026-07-28","io.modelcontextprotocol/clientCapabilities":
{},"io.modelcontextprotocol/clientInfo":{"name":"agentsql-smoke","version":"1.2.0"}}}}'
```

v1.2.0 可选的 MCP OAuth 入口

`/mcp/oauth` 是独立的 MCP `2026-07-28` 受保护资源入口，默认关闭。启用前先配置控制台 OIDC 身份源，并在同一租户创建对应的受管 OIDC 用户；`mcp.oauth.issuer_url` 和 `tenant_id` 必须与控制台 OIDC 配置完全一致。授权服务器须给该资源签发 RS256 JWT Access Token，其 `aud` 包含**精确的** `resource_url`，scope 包含配置的单个 `required_scope`。生产环境的 issuer 和 resource URL 必须使用 HTTPS；HTTP 仅用于本机回环测试。

```
mcp:
  oauth:
    enabled: true
    issuer_url: "https://id.example.com/realms/agentsql"
    resource_url: "https://mcp.example.com/mcp/oauth"
    tenant_id: "tenant_default"
    required_scope: "agentsql:mcp"
```

客户端预注册在授权服务器中，负责授权码、PKCE 和 `resource` 参数。MCP Inspector 2.11.0 的 `mcp.json` 可使用以下形式；其中 `scopes` 必须是字符串，Agent Key 与 OAuth Access Token 是两种独立凭据：

```
{
  "mcpServers": {
    "agentsql-oauth": {
      "type": "http",
      "url": "https://mcp.example.com/mcp/oauth",
      "protocolEra": "modern",
      "headers": {"X-AgentSQL-Agent-Key": "<Agent API Key>"},
      "oauth": {"clientId": "<预注册客户端 ID>", "scopes": "agentsql:mcp"}
    }
  }
}
```

未带 Access Token 时，服务器通过 `WWW-Authenticate` 指向 `/.well-known/oauth-protected-resource/mcp/oauth`。用户、Agent、租户和 `query.execute` 权限每次请求重新验证；控制台会话令牌、旧人员委托令牌、其他资源的令牌和 URL 查询参数都不能替代本入口的 Access Token。当前仅实测 Keycloak 26.4.6 + Inspector 2.11.0；Dify 1.17.1 只验证了旧 `/mcp` 入口。结果及边界见 v1.2.0 验收记录。

服务默认应只绑定回环地址。远程使用时，选择 SSH 本地转发或受控 TLS 反向代理；禁止让 `0.0.0.0:7780` 在公网裸奔。

```
# 适用前提：远端 AgentSQL 仅监听 127.0.0.1:7780；本机已配置 SSH 身份
ssh -N -L 7780:127.0.0.1:7780 user@agentsql-host
```

随后客户端仍连接 `http://127.0.0.1:7780/mcp`。使用反向代理时必须配置 TLS、访问控制和限流；不要记录 `Authorization` 头。

stdio：单机桌面 Agent

stdio 会启动一套新的 AgentSQL 运行时：

```
# 适用前提：本机已安装 agentsql；当前用户可读绝对路径配置及控制面存储
AGENTSQL_SECRET='<与控制面配对的恰好 32 字节值>' \
AGENTSQL_API_KEY='<Agent API Key>' \
agentsql mcp --config /absolute/path/config.yaml
```

`--config` 可缩写为 `-c`，默认是当前目录的 `config.yaml`。Key 优先通过 `AGENTSQL_API_KEY` 提供；不要把 Key 放进 `--api-key`，否则会出现在进程参数中。stdio 与 HTTP 使用同一套工具注册规则：普通 Agent Key 提供 12 个基础工具，默认开启 B5 时再提供 8 个会话/事务工具；任务级临时只读 Key 只提供 `query`。

> **重要：stdio 不是连接已运行 systemd 服务的代理。** 它会新建一套运行时，必须能读取配置、metadata/audit 控制面存储，并使用与加密数据源凭据相同的 `AGENTSQL_SECRET`。安装器生成的 `/etc/agentsql/agentsql.env` 权限为 `0600 root:root`，普通桌面用户读不到；已经安装并运行服务时请直接使用 HTTP。

stdio 模式会关闭控制台与事件流；不要期待它提供控制台 UI。

MCP 工具

所有 SQL 参数都只接受单条语句，不允许堆叠多语句。客户端必须读取结构化结果中的 `decision`。

普通 Agent Key 注册以下 12 个基础工具；任务级临时只读 Key 只注册 `query`：

工具	用途	入参	可能的 decision 与关键语义
`read_key`	读取 Redis/Valkey 指定单实例的精确键	`datasource_id`、`key`	只接受严格 TLS/ACL、独立 `key` 策略授权的 UTF-8 字符串；固定 `TYPE`/有界 `GETRANGE`，先脱敏和持久审计后交付，不接受通配符、原生命令、写入或跨库

工具	用途	入参	可能的 decision 与关键语义
`find_documents`	从指定 MongoDB 集合读取显式顶层标量字段	`datasource_id`、`collection`、`fields`、`filter`、`limit`	仅接受 1–8 个字符串等值过滤条件及 1–100 行上限；需要精确集合授权、严格 TLS、认证账号，返回前脱敏并写入持久审计；不接受聚合、任意操作符或写入
`query_analytics`	从指定 ClickHouse MergeTree 表读取显式列	`datasource_id`、`table`、`fields`、`filter`、`limit`	限 26.8.31.2 单实例、严格 TLS、只读账号、精确表与列授权、String 等值过滤及 1–100 行；v3.8.0 可选更严格读取预算，脱敏与持久审计成功后交付，不接受原生 SQL、视图或分布式表
`search_vectors`	在指定 Milvus collection 执行单向量相似度检索	`datasource_id`、`collection`、`vector_field`、`vector`、`fields`、可选 `filter`、`top_k`	限 2.6.25 TLS 单机、`default` 数据库、精确 collection 与字段授权、一个有限 FloatVector 和 1–20 个命中；v3.8.0 的管理员配置可开启权威租户范围及获准字段等值过滤；仅返回脱敏标量，不接受原生表达式、分区、写入或混合检索
`list_datasources`	列出当前 Agent 已获授权的数据源	无	成功返回公开标识信息，不返回 host、密码或连接参数；失败为 `error`
`list_schema`	读取已授权结构并按列权限裁剪	`datasource_id`；`table` 可选	成功通常为 `allow`；`table` 只接受字母、数字、下划线，以及至多一个点的 `table` 或 `schema.table`，不支持引号和特殊标识符；结果超过 10000 行时必须缩小到具体表
`explain_query`	解析并评估 SQL；已验证计划能力的方言附带执行计划，不执行查询	`datasource_id`、`sql`	可返回 `allow`、`deny`、`warn`、`approve` 或 `error`；即使 `allow` 也没有执行 SQL。Oracle Free 与崖山指定只读路径不预执行计划。
`query`	执行受保护的单条 `SELECT`	`datasource_id`、`sql`	仅查询；结果自动按数据源 `row_limit` 限行并做结果层脱敏；危险查询可 `deny` 或 `approve`
`execute_write`	提交单条 `INSERT`、`UPDATE`、`DELETE` 或 DDL	`datasource_id`、`sql`、必填 `reason`	管理类语句不走此工具；DDL 要求 Agent 为 `ddl`；`allow`/`warn` 才执行，`approve` 会自动创建 pending 审批单但不执行
`request_approval`	主动把 SQL 送审	`datasource_id`、`sql`、必填 `reason`	只创建审批单，不执行 SQL；返回 `approval_id`
`get_approval_result`	查询审批状态	`approval_id`	只能查询当前 Agent 自己的单；其他 Agent 的单号按未找到处理
`execute_approved`	执行已批准的封存请求	`approval_id`	原 Agent 单次认领；重新检查身份、数据源、策略和规则；拒绝重复调用

`execute_write` 和 `request_approval` 的调用方 `reason` 会作为申请说明保存；规则评估原因与审批备注单独保存。

v0.4.0 的 B5 跨请求逻辑会话与 PostgreSQL 计划事务出厂默认开启，因此默认 `tools/list` 还会返回以下 8 个工具：

工具	用途
`open_session` / `close_session` / `get_session_status`	创建、关闭和查询显式 AgentSQL 逻辑会话；不要把它与 HTTP `Mcp-Session-Id` 传输会话混用
`begin_transaction`	预检并封存完整有序 DML 计划，然后开始 PostgreSQL 事务；此步不执行语句
`execute_transaction_statement`	按计划 ordinal 执行一条已封存的 `INSERT`、`UPDATE` 或 `DELETE`，执行阶段不能替换 SQL
`commit_transaction` / `rollback_transaction`	提交或回滚并完成终态栅栏
`get_transaction_status`	幂等查询事务 outcome、audit 与 disposition 状态

B5 每个 operation 仍只允许一条顶层 SQL，不支持事务内 `SELECT`、`RETURNING`、DDL 或 MySQL 跨请求事务。显式设置 `mcp.sessions.enabled: false` 会隐藏这 8 个工具，并要求同时关闭 `mcp.transactions.postgres`；实际可用集合始终以 `tools/list` 为准。

调用时序

只读查询

```
list_datasources
  → list_schema
  → explain_query
  → query
```

先确认数据源和可见结构，再评估，最后执行。`explain_query` 的 `allow` 只说明评估通过，不等于已经查询。

写操作：自动转审批

```
execute_write(reason 必填)
  → decision="approve" + approval_id（系统已自动创建 pending 单）
  → 管理员在控制台审批
  → get_approval_result
  → approved 后原 Agent 调用 execute_approved(approval_id)
```

收到 `execute_write` 返回的 `approval_id` 后，不要再调用 `request_approval`，否则会重复送审。

`approved` 不会自动执行原 SQL。原 Agent 可在有效期内调用一次 `execute_approved`；该工具重新检查当前权限和规则，遇到拒绝或故障会关闭执行。旧审批单不可执行，单次认领后不得自动重试。

写操作：主动送审

```
request_approval(reason 必填)
  → approval_id
  → 管理员在控制台审批
  → get_approval_result
  → approved 后调用 execute_approved(approval_id)
```

主动送审本身从不执行 SQL；仅 `execute_approved` 在批准后执行封存请求。

正确处理返回值

- `decision:"allow"`：允许，执行类工具已经按其语义执行。
- `decision:"warn"`：成功返回并带告警；执行类工具可已执行，仍需记录警告。
- `decision:"deny"`：策略或规则拒绝；是成功返回的工具结果，不是传输错误。
- `decision:"approve"`：需要人工处理；SQL 没有执行。
- `decision:"error"`：处理失败；这是唯一会设置 MCP `IsError=true` 的决策。

因此，不要把 `deny`、`warn` 或 `approve` 当成网络错误自动重试。特别是写操作，盲目重试会产生重复审批或重复业务意图。

Claude Desktop 接入

Claude Desktop 使用 `claude_desktop_config.json` 中的 `mcpServers` 启动本机进程。v1.0.0 的 `agentsql mcp` 默认使用标准 MCP stdio 逐行 JSON 消息格式；旧版自定义长度前缀客户端可显式加 `--sealed-stdio` 继续使用原帧格式。以下直启示例适用于能运行 AgentSQL Linux 二进制的桌面环境；本版本发行资产的目标平台为 Linux amd64/arm64，实际下载须以完成验收的 Release 为准，Windows 桌面请使用下面的 HTTP 桥接示例。`command` 建议填写可执行文件的绝对路径，避免桌面应用与终端的 `PATH` 不一致。

```
{
  "mcpServers": {
    "agentsql": {
      "command": "/absolute/path/to/agentsql",
      "args": ["mcp", "--config", "/absolute/path/config.yaml"],
      "env": {
        "AGENTSQL_SECRET": "<与控制面配对的恰好 32 字节值>",
        "AGENTSQL_API_KEY": "<Agent API Key>"
      }
    }
  }
}
```

上述参数已经由仓库 CLI 实现确认：子命令是 `mcp`，配置参数是 `--config`（可缩写为 `-c`）。保存配置并重启 Claude Desktop 后，按工具注册规则默认应能看到 12 个基础工具和 8 个 B5 工具（本版未在 Claude Desktop 中实测，真实客户端验收使用 MCP Inspector 与 Dify）；显式关闭 B5 时只显示基础工具。若没有出现，先确认桌面客户端启动的进程能够读取可执行文件、配置文件、metadata/audit 控制面存储，并使用与加密数据源凭据相同的 `AGENTSQL_SECRET`。

Windows Claude Desktop：连接已有 AgentSQL 服务

先在本机或 SSH 本地转发端口运行 AgentSQL HTTP 服务，确认 `http://127.0.0.1:7780/healthz` 正常。Windows 客户端可用 □1□ 把本地 stdio 转为 Streamable HTTP；它是独立的社区桥接程序，不包含在 AgentSQL 发布包中。下例使用已安装的 `mcp-remote`；配置文件中的 `command` 请改成实际的 Node.js 可执行文件绝对路径，`args` 第一个值请改成实际 `mcp-remote` 入口脚本的绝对路径。Windows JSON 路径中的反斜杠需写成 `\\`。

```
{
  "mcpServers": {
    "agentsql": {
      "command": "C:\\path\\to\\node.exe",
      "args": [
        "D:\\path\\to\\mcp-remote\\dist\\proxy.js",
        "http://127.0.0.1:7780/mcp",
        "--transport", "http-only",
        "--header", "Authorization:${AGENTSQL_AUTH_HEADER}"
      ],
      "env": { "AGENTSQL_AUTH_HEADER": "Bearer <Agent API Key>" }
    }
  }
}
```

这里只给桌面桥接进程一个 Agent API Key。不要将管理员 token 用于 MCP；不要把监听端口暴露公网。Claude 的云端自定义连接器从 Anthropic 云网络发起连接，无法使用这里的回环地址；本例是 Claude Desktop 本地进程。桌面客户端需要完成自身登录后才能在界面查看工具，本仓库的协议测试不等于已登录客户端的 UI 验收。

MCP Inspector 本地验收（Streamable HTTP）

旧 `/mcp` 入口支持 MCP `2025-06-18`。该入口验收过的组合为 MCP Inspector `0.16.8` 与其依赖 `@modelcontextprotocol/sdk` `1.18.0`；较新的 Inspector 默认协商更新协议，不能直接套用这组旧入口验收步骤。在独立目录保存以下 `package.json`，运行 `npm install`，再启动本地 `node_modules/.bin/mcp-inspector`（Windows PowerShell 用 `./node_modules/.bin/mcp-inspector.cmd`）：

```
{
  "private": true,
  "dependencies": {"@modelcontextprotocol/inspector": "0.16.8"},
  "overrides": {"@modelcontextprotocol/sdk": "1.18.0"}
}
```

在 Windows PowerShell 中先设置 `$env:HOST='127.0.0.1'` 和 `$env:ALLOWED_ORIGINS='http://127.0.0.1:6274'`，再启动 Inspector，按终端给出的本地 URL 打开界面。传输选择 Streamable HTTP，地址填 `http://127.0.0.1:7780/mcp`，在 Authentication → Custom Headers 中填写完整的 `Authorization: Bearer <Agent API Key>`。不要再手工添加 `MCP-Protocol-Version`，该 SDK 在初始化后会自动发送；重复添加会形成无效的合并头。依次检查工具发现、授权查询、越权拒绝及对应审计 ID。使用单独的测试 Agent Key，不要填管理员 token，也不要把 Inspector 代理端口暴露到公网。Inspector 是 MCP 开发测试工具，不能替代 AI 助手或 Claude Desktop 的界面兼容性验收。

Dify 自托管接入（Streamable HTTP）

在 Dify 的 MCP 工具管理界面新增服务，Server URL 填 Dify API 容器可达的 `http://<AgentSQL 内网地址>:7780/mcp`，请求头同时填写 `Authorization: Bearer <Agent API Key>` 和 `MCP-Protocol-Version: 2025-06-18`。Dify 1.17.1 的 MCP 客户端声明该版本，但其 HTTP transport 不自动添加后续请求要求的版本头，因此此项必须显式配置。保存后先检查工具发现，再从工作流调用 `list_datasources` 和仅获授权的只读 `query`；返回 `deny`、`approve` 或 `error` 时按业务决策处理，不作为可盲重试的网络异常。Dify API 容器内的 `127.0.0.1` 指向 Dify 自己，不能指向宿主机 AgentSQL。

自托管 Dify 默认通过 SSRF 代理阻止访问私网地址；若 AgentSQL 位于同一受控内网，只在 Dify 的 `SSRF_PROXY_ALLOW_PRIVATE_IPS` 中加入 AgentSQL 实际地址的精确 `/32`，并保持 AgentSQL 对外端口受控。不要为接入测试放开全部 RFC1918 网段。Dify 的静态 MCP 请求头对应一个固定 Agent 身份；每位用户的独立 Agent 身份需要单独设计身份映射，不能把 Dify 工作区身份自动视为 AgentSQL 身份。

Cursor 接入（stdio）

Cursor 可把相同的 `mcpServers` 片段放入用户级 `~/.cursor/mcp.json`，或项目级 `.cursor/mcp.json`。团队项目优先只提交不含秘密的配置模板；真实 Key 和 SECRET 使用 Cursor 当前版本提供的秘密注入方式或仅保存在未纳入版本控制的本地配置中。

```
{
  "mcpServers": {
    "agentsql": {
      "command": "/absolute/path/to/agentsql",
      "args": ["mcp", "--config", "/absolute/path/config.yaml"],
      "env": {
        "AGENTSQL_SECRET": "<与控制面配对的恰好 32 字节值>",
        "AGENTSQL_API_KEY": "<Agent API Key>"
      }
    }
  }
}
```

项目级文件中的相对路径容易受工作目录影响，`command` 与 `--config` 的值均建议使用绝对路径。保存后按当前 Cursor 版本的方式刷新 MCP Server；若客户端版本的配置入口或外层字段发生变化，以 Cursor 官方 MCP 文档为准。

豆包及其他远程 MCP 客户端（Streamable HTTP）

客户端必须明确支持 **Streamable HTTP**，并允许为 MCP Server 添加自定义 Header。在豆包或其他客户端的 MCP Server 配置页填写：

配置项	公网演示	生产自托管
Server URL	`https://demo.agentsql.cn/mcp`	`https://<你的网关>/mcp`
Header 名	`Authorization`	`Authorization`
Header 值	`Bearer <演示环境 Agent API Key>`	`Bearer <自托管 Agent API Key>`

演示 Key 是 Agent API Key，不是演示控制台的管理员 token；使用登录页预填的演示管理员账号时，应确认所选演示 Agent 对应的当前只读 Key。若演示 Key 不公开，则改用自托管环境验证，不要猜测或复用管理员 token。演示环境仅限本地接入测试，生产必须自托管并使用自己的密钥。若某客户端不支持自定义 Header、只支持旧版 SSE，或无法确认其 Streamable HTTP 版本，则不能按此方式直连；不要把 Key 放入 URL 查询参数。

下面是通用字段骨架，豆包及其他客户端的配置入口、外层 JSON 结构，以及某些版本要求的 `type` 或 `transport` 字段，以该客户端当前官方文档为准：

```
{
  "mcpServers": {
    "agentsql": {
      "url": "https://demo.agentsql.cn/mcp",
      "headers": {
        "Authorization": "Bearer <Agent API Key>"
      }
    }
  }
}
```

已运行本机 AgentSQL 服务时，也可把 URL 换成 `http://127.0.0.1:7780/mcp`。非本机生产部署必须使用受控的 HTTPS 入口，不得把监听端口直接裸露到公网。

- Cursor：远程接入时，使用该版本官方 MCP 文档确认 HTTP transport 字段。
- Claude Desktop：使用官方 MCP 文档确认该版本是否支持 Streamable HTTP；若只支持 stdio，使用前面的 Claude Desktop 配置。
- Cline：使用官方 MCP 文档确认配置入口、`mcpServers` 结构及 HTTP transport 字段。

这些客户端的 `command`、`args`、`env`（stdio）和 `url`、`headers`（HTTP）是 AgentSQL 对接所需信息；外层结构与 transport 声明以客户端当前版本为准。

从 MCP Registry / Glama 查找

AgentSQL 已收录于 MCP 官方 Registry（状态为 active）。在 MCP Registry 搜索 **AgentSQL**，核对条目名称 `io.github.cuipengdba/agentsql`、版本和仓库地址后，再按客户端支持的方式接入；仓库没有给出可长期依赖的 Registry 条目直达 URL，因此本文不编造链接。

Glama 也已收录 AgentSQL，可在 Glama 的 AgentSQL 搜索结果 中查找。第三方目录中的端点、版本或 Header 示例可能滞后，最终以本仓库文档、实际部署配置和 MCP `tools/list` 返回为准。

客户端接入安全提醒

- 公网 demo 只连接合成演示库，用于功能体验和客户端连通性测试；不要向 demo 发送真实业务 SQL、数据或凭据。
- 生产环境必须自托管 AgentSQL，并使用自己的数据源、Agent 与 API Key；网关前配置 TLS、访问控制和限流。
- API Key、`AGENTSQL_SECRET` 和数据源口令不得写进公开配置、截图、日志或代码仓库。公开项目中只保留占位符，并在泄露后立即轮换。
- stdio 配置会在本机新建 AgentSQL 运行时；若已有受控服务，优先连接其 Streamable HTTP 端点，避免复制或放宽生产秘密文件权限。

- 客户端仅应获得完成任务所需的最小 Agent 能力和数据源授权；始终处理 `deny`、`warn`、`approve`、`error`，不要把安全决策当作网络失败自动重试。

自研客户端

initialize 裸 HTTP 示例

先设置 Key；以下请求适用于本机已运行的 Streamable HTTP 服务：

```
# 适用前提：本机 AgentSQL 已运行；AGENTSQL_API_KEY 已导出为有效 Agent Key
curl -fsS -D /tmp/agentsql-mcp-headers http://127.0.0.1:7780/mcp \
  -H "Authorization: Bearer ${AGENTSQL_API_KEY}" \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json, text/event-stream' \
  --data '{"jsonrpc":"2.0","id":1,"method":"initialize","params":{"protocolVersion":"2025-06-
18","capabilities":{},"clientInfo":{"name":"agentsql-smoke","version":"1.0.0"}}}'
```

Streamable HTTP 传输默认是有状态的。读取 initialize 响应中的 `Mcp-Session-Id`，并在后续 `tools/list`、`tools/call`、GET 和 DELETE 请求中原样携带。首次 initialize 通过 `params.protocolVersion` 协商版本，因此其 HTTP 协议版本头可省略；后续 POST 必须发送 `MCP-Protocol-Version: 2025-06-18`。服务端仍会对每个请求重新校验 Bearer。`mcp.http.stateful: false` 只用于兼容性回退，此时不会返回会话头且只接受 POST。无论哪种模式，每个请求都要发送 Bearer 和正确的 `Accept`，并把请求体控制在 4 MiB 内。

Go SDK

建议使用 `github.com/modelcontextprotocol/go-sdk` 的 `StreamableClientTransport`，通过自定义 `http.RoundTripper` 为每次请求注入：

```
Authorization: Bearer <Agent API Key>
MCP-Protocol-Version: 2025-06-18
Accept: application/json, text/event-stream
```

不要在日志中输出完整 Header，也不要把 `deny` 当作 transport error。

故障排查

`401 unauthorized`

检查是否误用了管理员 token，或 Agent Key 是否错误。Agent 被禁用、过期、删除、轮换后，旧 Key 都会失效；轮换不会保留并行有效期。

stdio 没有控制台 UI

这是预期行为。stdio 会关闭 `console_enabled` 与事件流。已安装服务应使用 HTTP，让控制台继续由 systemd 运行实例提供。

只能在服务器本机访问

默认回环绑定是安全设计。使用 SSH `-L`，或在受控 TLS 反代后访问；不要改成公网 `0.0.0.0` 裸奔。

MCP 握手失败

确认端点是 `/mcp`，请求包含 `Content-Type: application/json`、`Accept: application/json, text/event-stream` 和 `MCP-Protocol-Version: 2025-06-18`。默认有状态模式下还要确认 initialize 响应包含 `Mcp-Session-Id`，且后续请求携带相同的值；同时检查客户端是否真的支持 Streamable HTTP，以及请求体是否超过 4 MiB。

新入口 `/mcp/2026-07-28` 无需握手，每次请求必须携带版本 Header 和 `params._meta`；`server/discover` 只广告 `2026-07-28`。缺失、旧版或互相矛盾的版本元数据会返回 HTTP 400，超大请求体返回 413，跨站 Origin 返回 403。更多协议拒绝与升级排查见 当前版本运维手册。

堆叠多语句被拒绝

这是 R001 的预期行为。一次只发送一条 SQL；不要在合法查询后附加第二条语句。

写操作提示缺少 `reason`

`execute_write` 与 `request_approval` 都要求非空 `reason`。这类工具层前置拒绝发生在进入流水线之前，可能没有审计记录；补充目的、影响范围和回滚方案后重新提交。

stdio 连不上控制面或无法解密数据源

检查进程是否能读配置和 SQLite/PG 控制面，是否使用与数据源密文配对的同一个 `AGENTSQL_SECRET`。安装器的 `/etc/agentsql/agentsql.env` 是 `0600 root:root`；普通桌面用户读不到时不要复制或放宽生产秘密文件，改用已运行服务的 HTTP 端点。

SQLBot

v3.2.0 的 v2 适配器把 SQLBot v1.10.2 普通登录用户、工作空间和数据源逐条映射到 AgentSQL 人员与只读 Agent。管理员可在控制台配置绑定、导出不含密钥的配置、轮换连接凭据并查看配置诊断。可信 SQLBot 服务端提供当前用户身份，每次调用重新验证映射与当前权限；没有绑定或身份时拒绝执行。

本版个人绑定覆盖指定 PostgreSQL 和 MySQL 只读场景、固定源码哈希和受控 SQL 子集，包含模型元数据过滤、结构化拒绝定位及用户主动发起的只读审批。MySQL 的发现 schema 必须等于数据源数据库名；SQLBot 自己提供 MCP Server 不代表它默认调用 AgentSQL。安装、独立 Agent Key、TLS、审批操作页及旧 v1 配置的迁移边界见 SQLBot 接入指南。