

KI

MCP-Server für ImmoGenio: Hausverwaltung per KI- Agent steuern — 58 Tools über die Partner-API

Der ImmoGenio MCP-Server öffnet die Partner-API für KI-Agenten: 58 Tools über Objekte, Einheiten, Mieter, Verträge und Eigentümer, GenioFlow-Workflows statt CRUD, scope- und RLS-gesichert nach DSGVO.

AUTOR

ImmoGenio

VERÖFFENTLICHT

3. Juni 2026

ONLINE

www.immogenio.de/blog

Inhalt

01 Was MCP ist

02 Was der ImmoGenio-MCP-Server ist

03 Workflow statt CRUD — der eigentliche Unterschied

04 Sicherheit und DSGVO sind nicht optional

05 Einrichtung in wenigen Schritten

06 Grenzen der ersten Version

07 Fazit

EIN VERWALTER SITZT VOR EINER EINFACHEN FRAGE: „WELCHE OFFENEN AUFGABEN GIBT ES ZU OBJEKT 1000, UND WELCHER BESCHLUSS AUS DER LETZTEN VERSAMMLUNG WARTET NOCH AUF UMSETZUNG?“ Klassisch bedeutet das: zwei Module öffnen, zwei Filter setzen, zwei Listen abgleichen. Mit dem neuen ImmoGenio-MCP-Server stellt er dieselbe Frage in natürlicher Sprache an einen KI-Client – und der ruft im Hintergrund die passenden Tools auf, liest die Daten über die Partner-API und formuliert die Antwort. Kein Klickpfad, keine Maske, keine Schulung.

Dieser Beitrag erklärt, was der MCP-Server ist, was er kann, wo seine Grenzen liegen – und warum er weit mehr ist als eine Sprach-Hülle über einer Datenbank.

Was MCP ist

Das Model Context Protocol (MCP) ist ein offener Standard, über den KI-Anwendungen mit externen Systemen sprechen. Statt einem Sprachmodell eine API-Dokumentation vorzulegen und auf das Beste zu hoffen, definiert ein MCP-Server eine Menge klar beschriebener **Tools** – jedes mit Namen, Zweck und einem Eingabe-Schema. Der KI-Client (etwa Claude Desktop oder Claude Code) liest diese Tools beim Verbinden ein und entscheidet pro Anfrage, welches Tool mit welchen Parametern aufzurufen ist. Das Modell erfindet keine Endpunkte; es wählt aus einem festen, geprüften Katalog.

Für die Immobilienverwaltung ist das die entscheidende Eigenschaft: Die Reichweite eines KI-Agenten ist exakt das, was der MCP-Server freigibt – nicht mehr und nicht weniger.

Was der ImmoGenio-MCP-Server ist

Der MCP-Server (`@immogenio/mcp`) ist bewusst ein **dünnere Adapter**. Er enthält keine Geschäftslogik und keinen direkten Datenbankzugriff. Jedes Tool ruft ausschließlich einen Endpunkt der ImmoGenio-Partner-API v1 (`/api/v1/partner/*`) auf. Authentifizierung, Berechtigungsprüfung, Mandantentrennung, Pagination und Audit erzwingt vollständig die API – anhand eines Partner-API-Tokens. Der MCP-Server ist also kein zweiter Weg in die Daten mit eigenen Regeln, sondern dieselbe Tür wie das Portal, nur für Maschinen geöffnet. Wer die Idee einer offenen, aber kontrollierten Schnittstelle vertiefen möchte, findet sie im Beitrag [Offene API-Schnittstellen im Ökosystem](#).

Insgesamt stellt der Server **58 Tools** bereit, gruppiert um die fünf Kern-Entitäten der Verwaltung sowie die WEG-Vorgänge:

BEREICH	BEISPIEL-TOOLS
Objekte	<code>objekte_auflisten</code> (Filter Stadt/Baujahr, Aggregate zu Einheiten, Leerstand, aktiven Verträgen), <code>objekt_details</code> , <code>objekt_einheiten</code> , <code>objekt_eigentuemer</code>
Einheiten	<code>einheiten_auflisten</code> (Filter Typ, Status, Fläche, Zimmer), <code>einheit_details</code> und CRUD
Mieter	<code>mieter_auflisten</code> , <code>mieter_details</code> , <code>mieter_dokumente_auflisten</code> und CRUD
Mietverträge	<code>mietvertraege_auflisten</code> , <code>mietvertrag_details</code> , <code>mietvertrag_workflow_status</code> , <code>mietvertrag_workflow_transition</code>
Eigentümer	Stammdaten, Objekt- und Einheiten-Zuordnungen mit Anteilen, Mitglieder und Vertreter
Aufgaben	<code>aufgaben_auflisten</code> (Filter Status, Objekt, Fälligkeit)
Versammlungen / Beschlüsse	Auflisten, Details, CRUD sowie <code>..._workflow_status</code> und <code>..._workflow_transition</code>

Schreibende Tools rufen dabei nicht etwa eigene Logik im Partner-Endpunkt auf, sondern dieselben geteilten Domain-Services wie die normalen Portal-Routen – inklusive Start der GenioFlow-Workflows und der automatischen Folge-Übergänge. Eine über die API angelegte Eigentümerversammlung durchläuft denselben Prozess wie eine im Portal angelegte.

Workflow statt CRUD – der eigentliche Unterschied

Die meisten Schnittstellen behandeln Status als Datenfeld: Man setzt `status = "gekündigt"` und hofft, dass die Begleitprozesse irgendwo nachgezogen werden. ImmoGenio macht das anders. Vorgänge wie Kündigung, Beschlussfassung oder die Durchführung einer Versammlung sind endliche Zustandsmaschinen in der GenioFlow-Engine. Wie diese Engine arbeitet – Zustände, Guards, Side-Effects und Audit – beschreibt der Beitrag [GenioFlow: die Evolution der Immobilienverwaltung](#) ausführlich.

Der MCP-Server bildet genau dieses Prinzip ab. Statt eines Status-Feldes gibt es zwei Tools pro Vorgang:

- `*_workflow_status` liefert den aktuellen Zustand, ob er terminal ist, und vor allem die **erlaubten nächsten Aktionen** samt benötigtem Payload-Schema.
- `*_workflow_transition` führt **genau eine** dieser Aktionen aus – über den offiziellen `runWorkflowTransition`-Pfad, ohne Umgehung der Vorbedingungen.

Ein KI-Agent fragt also zuerst, was überhaupt möglich ist, und handelt dann gezielt. Drei Beispiele:

- **Mietvertrag:** Eine eingehende Kündigung wird mit dem Ereignis `kuendigung_eingegangen` angestoßen – der Vertrag tritt in den Kündigungs-Lebenszyklus ein, statt dass nur ein Feld umgesetzt wird. Die gestaffelten Fristen nach § 573c BGB bildet der Workflow serverseitig ab.
- **Beschluss:** Ein WEG-Beschluss wandert von `vorgeschlagen` über die Aufnahme in eine Versammlung zur Abstimmung. Beim Ereignis `annehmen` oder `ablehnen` werden Ja-Stimmen, Nein-Stimmen und Enthaltungen als Payload übergeben und atomar gespeichert; danach kann der Pfad über `umsetzung_starten` bis `umgesetzt` weiterlaufen oder eine erfolgreiche Anfechtung den Beschluss auf `ungueltig` setzen. Die rechtliche Grundlage der Stimmengewichtung behandelt [ETV-Voting nach § 23 WEG mit MEA-Gewichtung](#).
- **Eigentümerversammlung:** Eine Versammlung läuft von `geplant` über Tagesordnung, Einladung und Durchführung bis zum Protokollversand. Wie dieser digitale Ablauf im Portal aussieht, zeigt [Die digitale Eigentümerversammlung](#).

Damit kann ein KI-Agent reale Verwaltungsprozesse nicht nur lesen, sondern korrekt durchfahren – mit denselben Guards, Side-Effects und Protokollen, die auch im Portal greifen. Wie Sichtbarkeit der Übergänge, Guards und Payload-Schemas technisch zusammenspielen, vertieft [GenioFlow-Engine: Guards, Sichtbarkeit, Payload-Schemas](#); das Zusammenspiel von Master- und Sub-Workflows beschreibt [Master- und Sub-Workflows im Vermietungs-Lebenszyklus](#).

Sicherheit und DSGVO sind nicht optional

Eine Schnittstelle, die einem Sprachmodell Zugriff auf Mieterdaten gibt, muss bei der Datenschutz-Frage überzeugen. Der ImmoGenio-MCP-Server tut das nicht durch Vertrauen in den Client, sondern durch Erzwingung in der API.

Token, Tenant und Scopes. Ein Lauf des Servers nutzt genau ein Partner-API-Token (`imp_...`). Dieses Token bestimmt den Mandanten und die erlaubten Scopes. Ein Token mit nur `Objekt:Read` kann keine Mieter lesen und nichts schreiben – unabhängig davon, was das Modell versucht. Die Logik dahinter entspricht dem Rollen- und Rechtemodell des Portals, beschrieben in [RBAC: Rollen und Berechtigungen](#).

Abgestufte Personenbezugsdaten. Der Basis-Scope `Mieter:Read` liefert nur Identifikation – Name, Status, Verknüpfungen. Kontaktdaten wie E-Mail, Telefon, Anschrift oder Geburtsdatum erfordern den separaten Scope `Mieter:ReadPII`, analog bei Eigentümern. Bank- und Steuerdaten (IBAN, BIC, Steuer-ID) werden grundsätzlich nie exponiert – weder gelesen noch geschrieben. So lässt sich ein Agent einrichten, der Auswertungen über Bestände erstellt, ohne je eine einzige E-Mail-Adresse zu sehen.

Mandantentrennung und Audit. Die Trennung der Mandanten erzwingt PostgreSQL über Row-Level-Security auf Datensatz-Ebene; ein falsch geratenes Token sieht schlicht keine fremden Daten. Wie dieses Modell aufgebaut ist, erklärt [Row-Level-Security und Mandantentrennung](#). Jede schreibende Aktion über die Partner-API wird zudem revisionssicher protokolliert – der unveränderliche Audit-Trail ist in [Audit-Trail: Mutationen revisionssicher protokollieren](#) beschrieben.

Das Grundmuster – ein Sprachmodell ruft eng umrissene Tools auf, während die Mandantentrennung serverseitig garantiert bleibt – kennt ImmoGenio bereits aus dem Mieter-Chatbot; die Parallelen behandelt [KI-Mieter-Chatbot: Tenant-Isolation und Tool-Calling](#).

Einrichtung in wenigen Schritten

Die Verbindung ist bewusst schlicht gehalten. Zunächst erzeugt ein Tenant-Administrator ein Partner-API-Token mit den gewünschten Scopes; das Token wird einmalig zurückgegeben und nur als Hash gespeichert:

```
curl -X POST https://api.immogenio.de/api/partners/api-tokens \
  -H "Authorization: Bearer <portal-jwt>" \
  -H "Content-Type: application/json" \
  -d '{ "name": "MCP (lokal)",
        "scopes": ["Objekt:Read", "Task:Read", "Mietvertrag:Read", "Mietvertrag:Tran",
                  "Beschluss:Read", "Beschluss:Update"],
        "rate_limit_per_min": 60 }'
```

Anschließend wird der Server im KI-Client registriert. In Claude Desktop genügt ein Eintrag in der Konfiguration:


```
{
  "mcpServers": {
    "immogenio": {
      "command": "pnpm",
      "args": ["exec", "tsx", "apps/immogenio-mcp/src/main.ts"],
      "env": {
        "IMMOGENIO_API_URL": "https://api.immogenio.de",
        "IMMOGENIO_API_TOKEN": "imp_..."
      }
    }
  }
}
```

In Claude Code übernimmt das ein einzelner Befehl (`claude mcp add immogenio ...`). Sobald der Client verbunden ist, stehen die 58 Tools bereit, und Anfragen lassen sich in natürlicher Sprache stellen.

Grenzen der ersten Version

Die erste Version ist klar umrissen. Der Server ist **single-tenant**: ein Lauf, ein Token, ein Mandant. Ein Token, das mehrere Mandanten überspannt, ist bewusst kein Bestandteil von v1. Einzelne Endpunkte sind zudem absichtlich zurückgestellt – etwa die Belege-Abfrage, deren Datenmodell noch überarbeitet wird, sowie die Intake-Aktionen für Mieterhöhung und Mieterwechsel, deren Logik zunächst sauber in geteilte Services überführt werden muss. Die Kündigung ist über den Mietvertrags-Workflow bereits vollständig abgedeckt.

Diese Zurückhaltung ist Absicht: Lieber ein kleinerer, vollständig abgesicherter Tool-Katalog als eine breite Fläche mit halbfertigen Pfaden.

Fazit

Der ImmoGenio-MCP-Server macht die Verwaltung für KI-Agenten zugänglich, ohne die Kontrolle aus der Hand zu geben. Er liest und schreibt über dieselben Services wie das Portal, treibt Vorgänge über echte GenioFlow-Workflows statt über lose Status-Felder, und erzwingt Berechtigungen, abgestufte Personenbezugsdaten und Mandantentrennung in der API – nicht im Sprachmodell. Damit wird aus „mit der KI über die Verwaltung reden“ ein belastbarer, nachvollziehbarer Arbeitsweg.

Fragen, Rückmeldungen oder eigene Erfahrungen mit der Partner-API? Wir freuen uns über Ihre Nachricht an kontakt@immogenio.de oder über unseren [RSS-Feed](#).

