

Data Express

Ein geführtes Tool zur Datenbereinigung und -analyse, entwickelt für kleine und mittlere Unternehmen (KMU), deren Datenarbeit noch in Excel stattfindet

Erstellt im September 2026 · Kernanwendung sowie MCP-/Natural-Language-Schicht gebaut und funktionsfähig; ein kleines öffentliches Beispiel ist separat auf der Portfolio-Website eingebettet

1. Zusammenfassung

Wofür es gedacht ist. Die meisten kleinen und mittleren Unternehmen (KMU) wickeln ihre laufenden Daten — Verkäufe, Lagerbestände, Personalunterlagen — noch immer über Tabellenkalkulationen ab, und die meisten Personen, die mit diesen Tabellen arbeiten, sind keine ausgebildeten Datenanalysten. Einen unsauberen Export zu bereinigen, fehlende Werte korrekt zu behandeln und ein Diagramm zu erstellen, das eine Entscheidung tatsächlich stützt, erfordert Urteilsvermögen, das ein Nicht-Analyst nicht zwangsläufig besitzt und dessen Einstellung sich für eine Routineaufgabe kaum rechtfertigen lässt. Data Express soll genau diese Lücke schließen: einem fachfremden Nutzer einen geführten Weg zur gleichen Ergebnisqualität wie bei einem Spezialisten bieten, in vergleichbarer Geschwindigkeit, ohne selbst einer werden zu müssen.

Was es ist. Data Express ist eine Anwendung zur Datenbereinigung und -visualisierung, die Luis gezielt für diese Zielgruppe entwickelt hat — Unternehmen, deren Datenarbeit noch in Excel stattfindet und die von Personen betrieben wird, deren Berufsbezeichnung nicht Datenanalyst lautet. Statt eines leeren Arbeitsbereichs, der Fachwissen voraussetzt, führt die Anwendung den Nutzer durch eine feste, sinnvolle Abfolge von Schritten — Daten prüfen, Fehlendes oder Falsches behandeln, Typen konvertieren, Werte bereinigen, Diagramme erstellen —, sodass die Qualität des Ergebnisses vom Design des Tools abhängt und nicht von der Ausbildung des Nutzers. Eine neuere Natural-Language-Schicht sitzt auf derselben Engine auf und erlaubt es, in einfacher Alltagssprache zu beschreiben, was gewünscht ist, statt den geführten Ablauf manuell zu bedienen.

Was es leistet. In der geführten Form lädt ein Nutzer eine Tabelle hoch und wird Schritt für Schritt zu einem bereinigten, analysebereiten Datensatz sowie einer Reihe von Diagrammen geführt, wobei jede Änderung protokolliert wird, sodass nichts unbemerkt oder unumkehrbar geschieht. Über die Natural-Language-Schicht lässt sich dasselbe Ergebnis erreichen, indem

das Ziel direkt beschrieben wird — „diese Datei öffnen, die fehlenden Werte korrigieren und den Umsatz pro Quartal als Diagramm darstellen“ —, was sich, da diese Schicht auf einer offenen, toolaufrufbaren Schnittstelle basiert, auch auf die Steuerung durch ein anderes automatisiertes System erstreckt, nicht nur auf eine Person, die eine Anfrage eingibt. Ein kleines Beispiel dieser Natural-Language-Schnittstelle mit festem Datensatz ist zudem als kostenlose Vorschau auf Luis' Portfolio-Website eingebettet, sodass Besucher die zugrunde liegende Idee in Aktion sehen können — wenngleich es bewusst nur einen schmalen Ausschnitt dessen zeigt, was die vollständige Anwendung leistet.

2. Technische Zusammenfassung

Was die Anwendung technisch leistet. Data Express trennt seine Datenverarbeitungslogik strikt von jeder Schnittstelle, die darauf zugreift. Ein zentrales Modul implementiert Laden, Null-Wert-Behandlung, Typkonvertierung, Suchen-und-Ersetzen, Diagrammerzeugung und Export auf Basis eines pandas DataFrame, und jede Oberfläche des Projekts — die geführte Webanwendung, die Natural-Language-Tool-Schicht, das kleine öffentliche Beispiel — ruft dieselben Funktionen auf, statt sie neu zu implementieren, sodass das Verhalten über alle Oberflächen hinweg konstruktionsbedingt identisch bleibt.

Wie es umgesetzt ist. Die primäre Anwendung ist eine Streamlit-Anwendung (Python), aufgebaut auf pandas für Transformationen und Plotly für Diagramme, organisiert als feste Abfolge, der ein fachfremder Nutzer folgen kann, ohne raten zu müssen, was als Nächstes zu tun ist. Dieselbe Kernlogik wird als sieben Tools über das Model Context Protocol (MCP) bereitgestellt — Dateien auflisten, eine Datei öffnen, eine spaltenbezogene Übersicht abrufen, Nullwerte behandeln, ein Diagramm erstellen, den bereinigten Datensatz speichern, zurücksetzen —, die jeder MCP-fähige Client direkt aufrufen kann; eine Bridge zu einem lokal laufenden, kostenlosen Open-Source-Modell (über Ollama) erlaubt es bereits heute, die gesamte Pipeline mit Anweisungen in einfacher Alltagssprache zu steuern, ganz ohne Cloud-Abhängigkeit. Ein kleiner FastAPI-Dienst nutzt separat vier dieser sieben Tools hinter Googles Gemini-Modell als öffentliches Beispiel mit festem Datensatz für dieselbe Natural-Language-Schnittstelle.

Wie es entstanden ist. Die zentrale Transformationslogik wurde einmal geschrieben und bewusst wiederverwendet: Die MCP-Tool-Schicht, die Ollama-Bridge und das öffentliche Beispiel greifen alle auf dieselben Funktionen zurück, die auch die geführte Anwendung direkt aufruft, statt parallele Implementierungen manuell synchron zu halten. Die geführte, schrittweise Struktur der Hauptanwendung war selbst eine bewusste Designentscheidung im Hinblick auf die Zielgruppe — jemanden ohne formale Ausbildung in Datenanalyse — und nicht eine vereinfachte Version eines „fortgeschritteneren“ Tools; die Natural-Language-Schicht

wurde später auf demselben Fundament ergänzt, nachdem die Kernlogik bereits bestand, und erweitert damit, wer (oder was) sie steuern kann, nicht wie sie im Inneren funktioniert.

3. Fachlicher Anwendungsbereich

Das adressierte Problem

Kleine und mittlere Unternehmen (KMU) haben selten einen eigenen Datenanalysten, generieren aber dieselben Arten unsauberer, realer Daten wie größere Unternehmen: inkonsistente Exporte, fehlende Werte, Spalten, die umstrukturiert werden müssen, bevor jemand darauf reagieren kann. Die üblichen Optionen sind begrenzt — einen Spezialisten für Routine-Bereinigungsarbeiten einstellen oder beauftragen, einen Nicht-Spezialisten die Arbeit manuell erledigen lassen, mit uneinheitlicher Qualität und ohne Nachweis, was geändert wurde, oder die Analyse ganz auslassen. Data Express ist darauf ausgelegt, diesen Kompromiss aufzulösen: ein geführter Ablauf, der einen Nicht-Spezialisten zu einem Ergebnis in Spezialistenqualität führt, in Spezialistengeschwindigkeit, ohne Spezialisten-Personal.

Für einen bestimmten Nutzer entwickelt, nicht für einen generischen

Die geführte, schrittweise Struktur ist die zentrale Designentscheidung des Produkts, keine bloße Oberflächenpräferenz. Jede Phase des Ablaufs zeigt nur die Entscheidungen, die für diese Phase relevant sind — was mit fehlenden Werten zu tun ist, wie eine Spalte typkonvertiert wird, welches Diagramm zu den Daten passt —, sodass ein Nutzer nicht bereits die richtige Reihenfolge der Operationen oder die richtige Terminologie kennen muss, um zu einem korrekten Ergebnis zu gelangen. Genau das macht das Tool für die eigentliche Zielgruppe nutzbar: Büropersonal, Unternehmer und operative Mitarbeiter, die Tabellenkalkulationen als Teil eines umfassenderen Aufgabenbereichs handhaben, nicht als ihre Spezialisierung.

Vom geführten Ablauf zur vollständigen Automatisierung

Die auf derselben Engine aufbauende Natural-Language-Schicht verändert, wer — oder was — überhaupt beteiligt sein muss. Eine Verwaltungsmitarbeiterin oder ein Verwaltungsmitarbeiter kann bereits heute eine Bereinigungs- oder Diagrammaufgabe in einfacher Alltagssprache beschreiben und korrekt ausführen lassen, ohne die spezifischen Schritte des geführten Ablaufs zu erlernen. Da dieselbe Schnittstelle als Standard-MCP-Tools bereitgestellt wird und nicht als Einzelintegration, ist sie ebenso von einem anderen automatisierten System erreichbar, das diese Tools direkt aufruft — was letztlich einem Datenaufbereitungsschritt erlaubt, innerhalb des bestehenden automatisierten Workflows eines Unternehmens abzulaufen, ganz ohne dass eine Person die geführte Oberfläche oder eine Chat-Schnittstelle bedient.

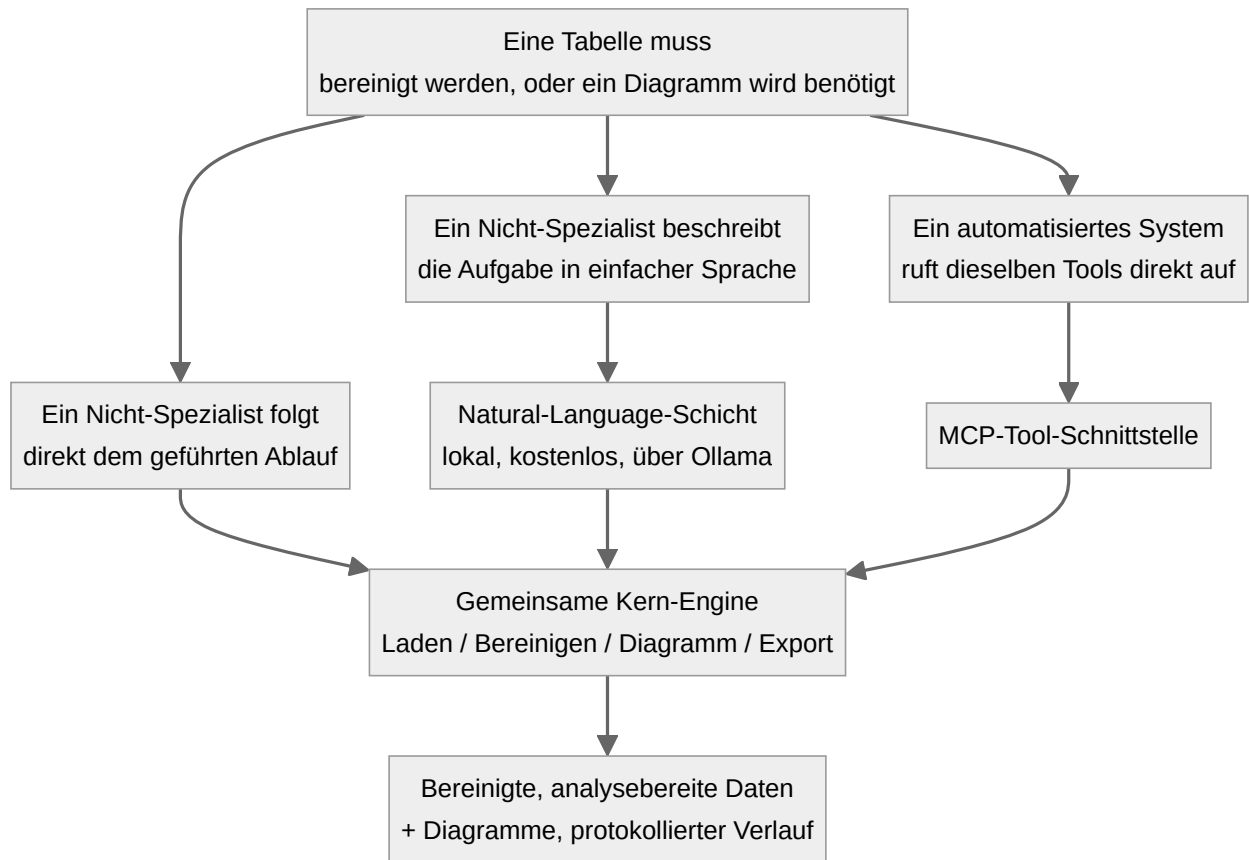


Abb. 1 — Drei Wege zum gleichen Ergebnis: manuell geführt, per Konversation geführt oder vollständig automatisiert.

Wer profitiert

ZIELGRUPPE	NUTZEN
Mitarbeitende in KMU (fachfremd)	Bereinigung und Analyse in Spezialistenqualität, ohne einen Spezialisten einzustellen
Unternehmer oder Führungskraft	Schnellere, konsistentere Datenaufbereitung mit einem sichtbaren Verlauf jeder vorgenommenen Änderung
Unternehmen, die Automatisierung wünschen	Dieselbe Engine ist über MCP direkt von anderen Systemen aufrufbar, ohne dass eine Person eine Oberfläche bedient

Das öffentliche Beispiel — und was es nicht ist

Eine kleine Version der Natural-Language-Schicht ist direkt in Envoys eigene Oberfläche eingebaut, als kostenlose Vorschau: ein fester Beispieldatensatz, vier sichere Operationen, kein Datei-Upload. Sie existiert einzig, um Besuchern zu ermöglichen, die zugrunde liegende Idee in weniger als einer Minute in Aktion zu sehen. Sie ist keine verkleinerte Version des Produkts und zeigt bewusst nur einen schmalen Ausschnitt dessen, was die geführte Anwendung und der vollständige Tool-Satz tatsächlich leisten — siehe Abschnitt 5 für die eigene Dokumentation.

4. Technischer Anwendungsbereich

Kernanwendung

Die Streamlit-App (`data-express-v2.1.py`) ist eine Ein-Datei-Python-Anwendung, aufgebaut auf pandas und Plotly, gegliedert in klare, geordnete Stufen, sodass ein Nicht-Spezialist immer weiß, welche Aktion als Nächstes möglich ist: Datei laden → Daten prüfen → Spalten löschen/umbenennen → Typen konvertieren → fehlende/ungültige Werte behandeln (der größte Abschnitt, da dies der Schritt ist, der am ehesten manuell uneinheitlich ausgeführt wird) → Suchen-und-Ersetzen → optionale Inline-Zellbearbeitung → speichern → bis zu zwölf Diagramme erstellen → Bereinigungsverlauf. Jede Transformation wird in einem sitzungsinternen Verlaufsprotokoll erfasst, sodass ein Nutzer — oder später ein Prüfer — genau nachvollziehen kann, was sich geändert hat und warum. Sie läuft eigenständig über Streamlit oder in Docker, auf Port 8501.

Natural-Language- und Automatisierungsschicht

Eine Headless-Version derselben Logik (`dataexpress_core.py`) bildet die Grundlage eines MCP-Servers, der sieben Tools über stdio bereitstellt: `list_files` , `open_file` , `get_overview` , `handle_nulls` , `create_chart` , `save_dataset` , `reset_data` . Da es sich um Standard-MCP-Tools handelt, kann jeder MCP-fähige Client die gesamte Pipeline steuern — dazu zählt heute bereits eine mitgelieferte Bridge zu einem lokal laufenden, kostenlosen Open-Source-Modell über Ollama, sodass eine Anweisung in einfacher Alltagssprache wie „diese Datei öffnen, alle Nullwerte löschen und die Einschreibungen pro Jahr als Diagramm darstellen“ ausgeführt wird, wobei jeder Tool-Aufruf und jedes Ergebnis zur Nachvollziehbarkeit angezeigt wird, ganz ohne Cloud-Kosten. Dieselbe Schnittstelle steht ebenso jedem anderen automatisierten System offen, was letztlich erlaubt, den Datenaufbereitungsschritt auszuführen, ohne dass eine Person die geführte Oberfläche oder eine Chat-Schnittstelle bedient.

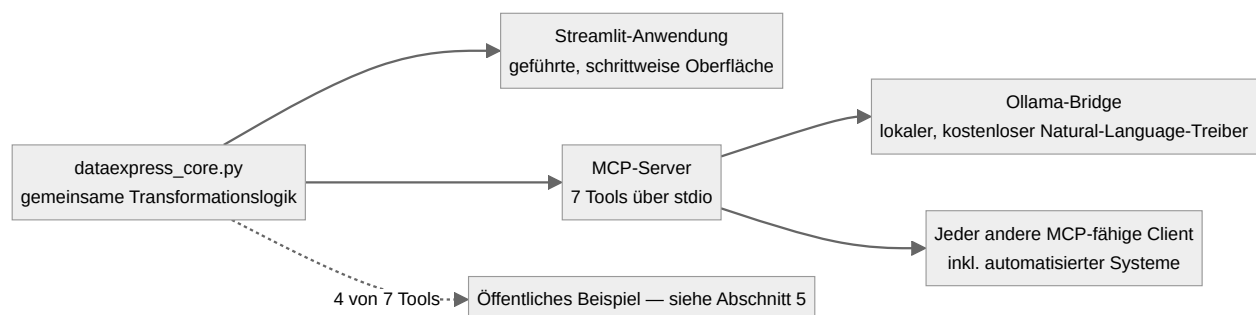


Abb. 2 — Eine gemeinsame Engine hinter der geführten Anwendung, der Natural-Language-/Automatisierungsschicht und dem öffentlichen Beispiel.

Öffentliches Beispiel

Ein kleiner FastAPI-Dienst nutzt vier der sieben Tools hinter Googles Gemini-Modell als öffentlich zugängliches Beispiel, begrenzt auf einen festen, sitzungsgebundenen In-Memory-Datensatz ohne Datei-Upload und ohne Schreibvorgänge auf die Festplatte. Sein Widget ist als natives Bauteil direkt in Envoys eigenes React-Frontend integriert (keine separat eingebettete Seite) und bietet sowohl einen manuellen Modus — direkte Aufrufe der Tool-Funktionen ohne Modell — als auch einen Agentenmodus mit demselben Gemini-Function-Calling-Ablauf, erreicht über Envoys eigenen Server, der zu diesem Backend proxyt. Es handelt sich um eine schmale Demonstration, nicht um eine Vorschau auf die tatsächliche Leistungsfähigkeit des Produkts — die vollständige Dokumentation befindet sich in Abschnitt 5.

5. Demo

Zusammenfassung

Das Chat-Widget auf Luis' Portfolio-Website ist ein kleines, kostenloses Beispiel der Natural-Language-Schnittstelle von Data Express — keine verkleinerte Version des Produkts. Es läuft gegen einen einzigen festen Datensatz und stellt vier sichere Operationen bereit, einzig damit ein Besucher in weniger als einer Minute sehen kann, wie die Natural-Language-Schicht im Prinzip funktioniert. Es spiegelt weder den vollständigen, schrittweisen Workflow der geführten Anwendung wider, noch den Umgang mit den echten Daten eines Unternehmens, noch die umfassendere Natural-Language- und Automatisierungsfähigkeit, die dieselbe Engine an anderer Stelle bereits unterstützt.

Technische Zusammenfassung

Technisch gesehen ist das Beispiel ein eigenständiger FastAPI-Dienst, der vier der sieben MCP-Tools von Data Express als Function-Calling-Tools für Googles Gemini-Modell bereitstellt, gegen eine In-Memory-Kopie eines festen Datensatzes, die auf die jeweilige Sitzung begrenzt ist — kein Datei-Upload, nichts wird auf die Festplatte geschrieben. Rate Limiting (20 Nachrichten pro 10 Minuten), eine Obergrenze für Tool-Aufrufe pro Runde, eine Begrenzung der Nachrichtenlänge sowie ein deterministischer, schlüsselwortbasierter Fallback (automatisch verwendet, wenn kein API-Schlüssel konfiguriert ist oder der Live-Aufruf fehlschlägt) halten es unabhängig von API-Kosten sicher und testbar. Es nutzt dieselben Kern-Transformationsfunktionen wie die geführte Anwendung, sodass sein Verhalten so weit korrekt ist, wie es reicht — es ist lediglich bewusst schmal gehalten, keine Vorschau auf die tatsächliche Tiefe des Produkts.

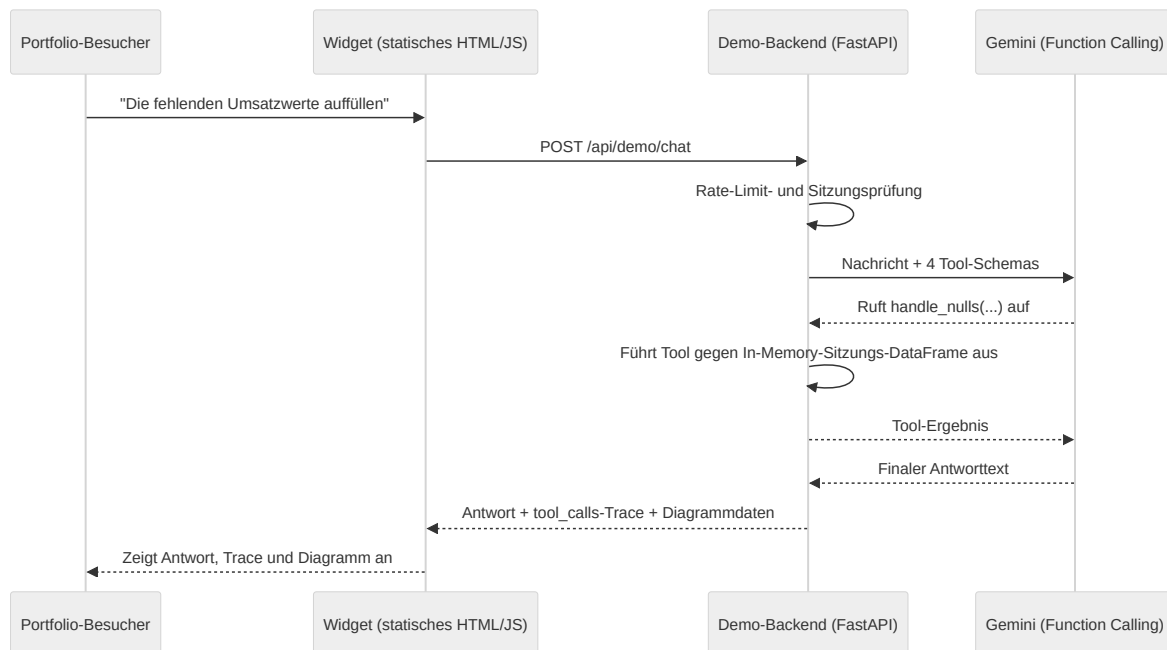


Abb. 3 — Ablauf der Demo-Anfrage: von der Besuchernachricht bis zum dargestellten Ergebnis, alles innerhalb des Vier-Tool-Beispiels.

Fachlicher und technischer Umfang

Als Portfolio-Stück ist es die Aufgabe des Beispiels, die zugrunde liegende Idee in wenigen Sekunden Interaktion greifbar zu machen, nicht das Produkt darzustellen. Die Begrenzung auf vier Operationen an einem festen Datensatz ist eine bewusste Scoping-Entscheidung für eine öffentliche, nicht authentifizierte Seite — keine Einschränkung der zugrunde liegenden Engine, die bereits den vollständigen Satz von sieben Tools jedem MCP-Client zur Verfügung stellt, einschließlich der lokal laufenden Natural-Language-Schnittstelle, die schon heute die volle Funktionalität der geführten Anwendung steuert.

Das Backend des Beispiels läuft als eigener Container, angebunden an das externe Netzwerk, das das Envoy-Projekt (CV-Website) erstellt, sodass Envoys Next.js-Server intern dorthin proxyen kann, ohne einen neuen öffentlichen Port zu öffnen. Sein Frontend ist ein nativer Port des eigenständigen Widgets, direkt in Envoys eigene React-Codebasis integriert, mit zwei Modi nebeneinander — manuell (direkte Tool-Aufrufe, kein Modell) und Agent (Gemini Function Calling) —, die beide auf demselben In-Memory-Sitzungsdatensatz arbeiten, sodass ein Besucher den Unterschied zwischen manuell und Agent direkt sehen kann, statt nur den Chat-Trick gezeigt zu bekommen. Es ist vollständig gebaut, funktionsfähig und in Envoys eigene Oberfläche eingebettet, keine ausstehende Integration.

