KI-Assistent einrichten

Der **KI-Assistent** ist ein Chat im Seitenpanel von Webdesk. Er beantwortet Fragen des angemeldeten Benutzers und kann dafür vordefinierte **Tools** aufrufen, mit denen er Daten aus Webdesk liest (Personen, Gruppen, Zeiterfassung, Projektzeit) und in einem Fall auch schreibt (Anwesenheitsbuchung). Die Antworten erzeugt ein externes Sprachmodell, das der Administrator über ein **Modell** (Zugangsdaten und Endpunkt) anbindet. Optional fassen **Agenten** Modell, Zusatz-Anweisungen und erlaubte Tools zusammen.

Die Seite richtet sich an Administratoren und Consultants und beschreibt die Einrichtung, die Reihenfolge, in der Schlüssel, Modell und Endpunkt ermittelt werden, den Werkzeugkatalog, die Berechtigungsprüfung und die Datenflüsse zum externen Anbieter. Eine kurze Erwähnung der Funktion findet sich in Neue Features[1]; die allgemeine Bedienung der Systemparameter ist in Systemparameter[2] beschrieben.

Achtung: Sobald der Assistent konfiguriert ist, werden Texte des Benutzers, Angaben zu seiner Person und die Ergebnisse der aufgerufenen Tools an den konfigurierten externen Anbieter übertragen (Details im Abschnitt *Datenschutz*). Vor der Aktivierung muss geklärt sein, ob dies für das Unternehmen zulässig ist.

Voraussetzungen

- **Lizenz:** Die Lizenzdefinition ai-impl.ldef.xml definiert die Lizenz ai (Modul ai) mit der Beschreibung "Anzahl der Benutzer, die mit benutzerdefinierten AI Agenten chatten dürfen". Die Lizenz ordnet dem Modul seine Spring-Kontexte und alle seine Aktionen zu (keine Aktion ist ausgeschlossen). Die Lizenzauslastung ist unter *System* > *Aktuelle Lizenz* einsehbar (siehe Lizenz-Information[3]).
- **Netzwerk:** Der Webdesk-Server muss den Endpunkt des Anbieters (Standard: https://api.openai.com/v1) per HTTP(S) erreichen können.
- **Zugangsdaten:** Ein API-Schlüssel des Anbieters, der Modellname (bei Azure OpenAI: der Deployment-Name) und bei anderen Anbietern als OpenAI die Basis-URL.
- **Berechtigungen:** Administratoren benötigen die Aktionen ai_showModels.act und ai_showAgents.act (Bearbeiten über ai_editModel.act bzw. ai_editAgent.act), Benutzer die Aktion ai_assistant.act. Die Berechtigungen werden wie bei jeder anderen Aktion vergeben.

Menü und Aktionen

Im Administrationsmenü gibt es unter *Module* > *AI Administration* drei Einträge:

Menüeintrag	Aktion	Zweck
AI Modelle	ai_showModels.act (Bearbeiten: ai_editModel.act)	Liste der Modelle (Spalten Name, Provider, Model ID, Basis URL, Mandant); Schaltfläche "Neues AI Modell".
AI Agenten	ai_showAgents.act (Bearbeiten: ai_editAgent.act)	Liste der Agenten (Spalten Name, Beschreibung, Model ID, Mandant); Schaltfläche "Neuer AI Agent".
Assistent	ai_assistant.act	Der Chat selbst (Beschreibung "AI Assistent Chat"). Zugriff auf die REST-Schnittstelle /rest/ai/... für GET und POST.

Die Listen zeigen Einträge der zugänglichen Mandanten sowie die *öffentlichen* Einträge (kein Mandant gesetzt, Anzeige "Öffentlich"). Der Link zum Bearbeiten öffentlicher Einträge erscheint nur für Benutzer, die die Daten aller Mandanten ansehen dürfen.

Symbol im Seitenpanel

In der Kopfzeile erscheint das Assistent-Symbol (Tooltip "Assistent") nur, wenn die Aktion ai_assistant.act existiert und der Benutzer dafür berechtigt ist. Ein Klick öffnet bzw. schließt das Seitenpanel, das sich anheften (ab einer Fensterbreite von 1200 Pixeln), minimieren und im Vollbild öffnen lässt. Fehlt das Symbol, fehlt die Berechtigung oder die Lizenz ai.

Einrichtung

1. **Modell anlegen:** *AI Modelle* > "Neues AI Modell" (Felder siehe nächster Abschnitt) und speichern.
2. **Optional Agent anlegen:** *AI Agenten* > "Neuer AI Agent", das Modell auswählen, bei Bedarf System Prompt und Tools festlegen.
3. **Alternativ globale Vorgaben setzen:** API-Schlüssel, Modell und Basis-URL als Umgebungsvariable, JVM-Parameter oder Systemparameter AiOptions hinterlegen (Reihenfolge siehe unten). Diese Vorgaben gelten, wenn kein Agent bzw. kein Modell verwendet wird, und ersetzen einen leeren API-Schlüssel im Modell.
4. **Berechtigung vergeben:** Den Benutzern bzw. Rollen die Aktion ai_assistant.act sowie die Aktionen der Module geben, deren Tools genutzt werden sollen (siehe Werkzeugkatalog).
5. **Testen:** Als Benutzer das Panel öffnen und eine Frage stellen, z.B. nach dem Resturlaub. Die Antworten auf Fehlerfälle stehen im Abschnitt *Fehlersuche*.

Modell ("AI Modell")

Ein Modell beschreibt, wie Webdesk ein Sprachmodell eines Anbieters anspricht. Pflichtfelder sind Name, Provider und Model ID.

Feld	Erklärung	mögliche Wert(e)
Name	Bezeichnung des Modells in Listen und Auswahldialogen.	Text, Pflichtfeld
Mandant	Bindet das Modell an einen Mandanten. Ohne Mandant ist das Modell öffentlich und steht allen Mandanten zur Verfügung. Ein Agent kann nur Modelle verwenden, die öffentlich sind oder zu seinem eigenen Mandanten gehören.	Mandant oder leer (Öffentlich)
Provider	Art der Schnittstelle. "OpenAI-kompatibel" (gespeichert als OpenAi) spricht die OpenAI-Chat-Schnittstelle an, also auch andere Dienste mit kompatibler Schnittstelle (auch über http://; es wird HTTP/1.1 verwendet). "Azure OpenAI" (gespeichert als AzureOpenAi) verwendet die Azure-Schnittstelle. Wird "OpenAI-kompatibel" gewählt, die Basis-URL aber enthält openai.azure.com oder /openai/deployments/, wechselt Webdesk automatisch auf Azure OpenAI.	OpenAI-kompatibel / Azure OpenAI, Pflichtfeld

Model ID	Kennung des Modells beim Anbieter. Bei Azure OpenAI ist es der Deployment-Name; enthält die Basis-URL einen Abschnitt /openai/deployments/<name>, hat dieser Name Vorrang vor der Model ID (Warnung im Log).	Text, Pflichtfeld, z.B. gpt-4o
Basis URL	Adresse des Dienstes. Leer: Es gilt die globale Vorgabe (siehe Reihenfolge), zuletzt https://api.openai.com/v1. Ein angehängtes /chat/completions wird entfernt. Bei Azure OpenAI ist die Basis-URL zwingend; die API-Version wird aus dem Parameter api-version der URL übernommen, und ein Host auf .cognitiveservices.azure.com wird auf .openai.azure.com umgeschrieben.	URL (Standard: leer)
Routing-Praeferenz	Wunsch an Routing-Dienste, der als Parameter preference mitgesendet wird. Wirkt nur, wenn die Basis-URL cortecs.ai enthält; für andere Endpunkte wird der Wert ignoriert (Warnung im Log). Bei Cortecs-Endpunkten sendet Webdesk zusätzlich include_metadata und schaltet parallele Tool-Aufrufe ab.	Standard (leer) / Ausgewogen (balanced) / Kostenoptimiert (cost) / Geschwindigkeitsoptimiert (speed)
API Schluessel	API-Schlüssel des Anbieters, Eingabe als Passwortfeld. Der Wert wird unmittelbar als Schlüssel verwendet (keine Referenz auf einen externen Tresor). Leer: Es gilt die globale Vorgabe (siehe Reihenfolge).	Text, z.B. <your-api-key>

Hinweis: Das Modellfeld "API Schluessel" hat im Code keine Verschlüsselung beim Speichern. Der Systemparameter AiOptions.defaultAiApiKey wird dagegen verschlüsselt gespeichert und in der Maske als Passwort dargestellt. Für Installationen mit einem einzigen Anbieter ist es daher sinnvoll, den Schlüssel dort (oder als Umgebungsvariable) zu hinterlegen und das Modellfeld leer zu lassen.

Agent ("AI Agent")

Ein Agent legt fest, mit welchem Modell der Assistent arbeitet, welche zusätzlichen Anweisungen er erhält und welche Tools er verwenden darf. Pflichtfelder sind Name und Modell.

Feld	Erklärung	mögliche Wert(e)
Name	Anzeigename. Er erscheint in der Agent-Auswahl des Chats	Text, Pflichtfeld

	und bestimmt die Reihenfolge (alphabetisch).	
Mandant	Bindet den Agenten an einen Mandanten. Ohne Mandant ist der Agent öffentlich. Ein Benutzer sieht und verwendet nur Agenten seines Mandanten und öffentliche Agenten. Der Auswahldialog für das Modell zeigt öffentliche Modelle und, wenn hier ein Mandant gewählt ist, die Modelle dieses Mandanten; ohne Mandant nur öffentliche Modelle.	Mandant oder leer (Öffentlich)
Beschreibung	Freitext zur Dokumentation (maximal 1024 Zeichen). Wird in der Agentenliste angezeigt, aber nicht an das Sprachmodell gesendet.	Text
Modell	Das zu verwendende "AI Modell". Gehört das Modell zu einem anderen Mandanten als dem des Benutzers, wird stattdessen die globale Vorgabe verwendet (Warnung im Log).	Auswahl, Pflichtfeld
System Prompt	Zusätzliche Anweisungen an das Sprachmodell (maximal 4000 Zeichen). Sie werden an den von Webdesk erzeugten Standard-Prompt angehängt und ersetzen ihn nicht.	Text
Erlaubte Tools	Zweispaltige Auswahl "Verfuegbare Tools" / "Ausgewaehlte Tools" mit allen im System registrierten Tools. Die Auswahl schränkt die Tools ein, die dem Modell angeboten werden. Bleibt die Auswahl leer, gilt keine Einschränkung: Es werden alle Tools angeboten, die der Benutzer nach seinen Berechtigungen verwenden darf. Gespeichert werden die Tool-Namen kommagetrennt (maximal 4000 Zeichen).	Tool-Namen, leer = alle erlaubten

Welcher Agent wird verwendet?

- Wählt der Benutzer in der Agent-Auswahl des Chats einen Agenten, wird dieser verwendet, sofern er öffentlich ist oder zum Mandanten des Benutzers gehört. Sonst weicht Webdesk auf den *Standard-Agenten* aus: den alphabetisch ersten Agenten des Mandanten, ersatzweise den alphabetisch ersten öffentlichen Agenten, ersatzweise den ersten zugänglichen Agenten.

- Bleibt die Auswahl bei "Default agent (auto)" (leerer Wert), verwendet Webdesk einen internen Standard-Assistenten ohne Modell und ohne Tool-Einschränkung. Modell, Basis-URL und Schlüssel stammen dann ausschließlich aus den globalen Vorgaben, und es stehen alle Tools zur Verfügung, die der Benutzer nach seinen Berechtigungen verwenden darf. Schlägt der Aufruf in diesem Modus mit "Modell nicht gefunden" fehl (HTTP 404), wiederholt Webdesk ihn einmal mit dem Standard-Agenten des Mandanten.

Achtung: Die Tool-Auswahl eines Agenten ist keine Sicherheitsgrenze. Jeder Benutzer kann in der Agent-Auswahl "Default agent (auto)" wählen, wodurch die Einschränkung des Agenten entfällt. Maßgeblich für die Sicherheit sind die Aktionsberechtigungen (siehe Berechtigungsprüfung).

Reihenfolge der Einstellungen (Fallback)

Webdesk ermittelt API-Schlüssel, Modellname und Basis-URL bei jeder Anfrage. Der erste nicht leere Wert der jeweiligen Liste gilt. Die Umgebungsvariablen und JVM-Parameter heißen unabhängig vom Anbieter OPENAI_API_KEY, OPENAI_MODEL und OPENAI_BASE_URL und haben Vorrang vor dem Systemparameter AiOptions.

Einstellung	Reihenfolge (erster nicht leerer Wert gewinnt)	Ergebnis, wenn nichts gesetzt ist
API-Schlüssel	1. Feld "API Schluessel" des Modells des Agenten 2. Umgebungsvariable OPENAI_API_KEY 3. JVM-Parameter -DOPENAI_API_KEY=<your-api-key> 4. Systemparameter AiOptions.defaultAiApiKey	Kein Aufruf des Anbieters. Der Chat antwortet "AI chat fallback is active; no API key configured."
Modellname	1. Feld "Model ID" des Modells des Agenten (bei Azure mit URL-Deployment: der Name aus der URL) 2. Umgebungsvariable OPENAI_MODEL 3. JVM-Parameter -DOPENAI_MODEL=... 4. Systemparameter AiOptions.defaultAiModel (Standard: gpt-4o) 5. fest eingebauter Wert gpt-4o	gpt-4o
Basis-URL	1. Feld "Basis URL" des Modells des Agenten 2. Umgebungsvariable OPENAI_BASE_URL 3. JVM-Parameter -DOPENAI_BASE_URL=... 4. Systemparameter AiOptions.defaultAiBaseUrl (Standard: leer)	https://api.openai.com/v1

	5. fest eingebauter Wert https://api.openai.com/v1	

Das Feld des Modells wird nur berücksichtigt, wenn der Chat einen Agenten mit zugänglichem Modell verwendet. In allen anderen Fällen ("Default agent (auto)", Modell eines fremden Mandanten) entfällt Schritt 1, und es wird mit dem Provider "OpenAI-kompatibel" und den globalen Vorgaben gearbeitet. Ist im Modell zwar ein Modellname, aber kein Schlüssel eingetragen, wird der Schlüssel nach Schritt 2 bis 4 bestimmt.

Systemparameter AiOptions

Die Bean AiOptions enthält die globalen Vorgaben. Sie gelten nur, wenn weder Modell noch Umgebungsvariable bzw. JVM-Parameter einen Wert liefern.

Eigenschaft	Erklärung	mögliche Wert(e)
defaultAiApiKey	Standard-API-Schlüssel als Fallback, wenn keine Umgebungsvariable und kein JVM-Parameter konfiguriert ist. Wird verschlüsselt gespeichert und in der Maske als Passwort dargestellt.	Text (Standard: leer)
defaultAiModel	Standard-Modellkennung als Fallback, wenn keine Umgebungsvariable und kein JVM-Parameter konfiguriert ist.	Text (Standard: gpt-4o)
defaultAiBaseUrl	Standard-Basis-URL als Fallback, wenn keine Umgebungsvariable und kein JVM-Parameter konfiguriert ist. Leer bedeutet https://api.openai.com/v1.	URL (Standard: leer)

Werkzeugkatalog (Tools)

Tools sind im Programm festgelegt und können nicht über die Oberfläche angelegt werden. Sie werden von den Modulen Portal & Organisation (PO), Zeiterfassung (TA) und Projektzeiterfassung (PTM) bereitgestellt. Alle Tools laufen im Namen des angemeldeten Benutzers: Die Tools "getMy..." liefern ausschließlich seine eigenen Daten, die übrigen Tools nur Daten, die er nach den unten beschriebenen Prüfungen sehen darf. Mit "Schreibend" sind Tools markiert, die Daten verändern.

Portal & Organisation

Tool	Zweck	Erforderliche Aktion (eine genügt)
searchColleagues	Sucht Kollegen im Mandanten nach Suchbegriff (Name, Mitarbeiter-ID, Kurzname, Organisationseinheit).	po_showPersons.act, po_searchPerson.act
findPersonsByName	Sucht Personen nach vollständigem oder teilweisem Namen.	po_showPersons.act, po_searchPerson.act
getAllGroups	Alle Gruppen des Mandanten (Organisationseinheiten,	po_showGroups.act

	Kostenstellen, Standorte, Abrechnungskreise).	
searchGroups	Sucht Gruppen nach Name oder Kurzname.	po_showGroups.act
getPersonsAge	Alter einer Person (Parameter personUid).	po_showPersons.act
getPersonsManager	Vorgesetzter einer Person (Standardrolle Manager).	po_showPersons.act
getPersonsCompetence	Personen im Zuständigkeitsbereich einer Person (auch über deren Gruppen).	po_showPersons.act
getPersonsGroups	Gruppen, denen eine Person zugeordnet ist.	po_showPersons.act, po_showGroups.act, po_searchPerson.act
getPersonsOfGroup	Mitglieder einer Gruppe (Parameter groupUid).	po_showGroups.act, po_showPersons.act, po_searchPerson.act

Zeiterfassung

Tool	Zweck	Erforderliche Aktion (eine genügt)
getMyOpenVacationDays	Offene Urlaubstage des laufenden Urlaubsjahres.	ta_getJournalV2.act
getMyYearlyVacationDaysEntitlement	Jahresurlaubsanspruch in Tagen.	ta_getJournalV2.act
getMyPreviousVacationIntervals	Urlaubsintervalle der letzten 12 Monate.	ta_getJournalV2.act, ta_doBooking.act
getMyWeeklyWorkingTime	In der laufenden Woche bereits gearbeitete Stunden.	ta_getJournalV2.act
getMyWeeklyTargetTime	Wöchentliche Soll-Stunden laut Zeitmodell.	ta_getJournalV2.act
getBookingsInTimeRange	Eigene Buchungen eines Tages (Parameter day).	ta_getJournalV2.act
getMyTodayBookings	Eigene Buchungen des heutigen Tages.	ta_getJournalV2.act
getAbsenceIntervals	Genehmigte vergangene und geplante eigene Abwesenheiten in einem Zeitraum (Parameter from, to).	ta_getJournalV2.act, ta_getCalendar.act, ta_yearlyCalendar.act
listUpcomingAbsences	Eigene genehmigte künftige Abwesenheiten ab heute (Parameter daysAhead, Standard 90, höchstens 3650 Tage).	ta_getJournalV2.act, ta_getCalendar.act, ta_yearlyCalendar.act
getTimeEntryReasons	Alle Zeitgründe des Mandanten, mit denen gebucht werden kann (Arbeit und Abwesenheit).	ta_doBooking.act, ta_getAttendanceTableau.act

getAttendanceStateOfColleague	Anwesenheitsstatus einer Person (abwesend, anwesend, abwesend mit Grund, bucht nicht) samt letzter Buchungsmeldung (Parameter personUid).	ta_getAttendanceTableau.act
doBooking (Schreibend)	Führt eine automatische Anwesenheitsbuchung (Kommen/ Gehen) für den Benutzer mit dem gewählten Zeitgrund durch (Parameter timeEntryReasonId). Nach erfolgreicher Buchung lädt die Seite im Hintergrund neu. Der Aufruf erfolgt ohne Rückfrage durch Webdesk.	ta_doBooking.act

Projektzeiterfassung

Tool	Zweck	Erforderliche Aktion (eine genügt)
getMyMonthlyEffortSummary	Monatsübersicht der eigenen Projektzeiten je Projekt (Parameter month, year).	ptm_editMonthlyTimeEfforts.act, ptm_editTimeEfforts.act
listMyOpenProjectAssignments	Projekte, auf die der Benutzer buchen und/oder Buchungen ändern darf.	ptm_editTimeEfforts.act, ptm_doProjectTimeBooking.act, ptm_editMonthlyTimeEfforts.act
searchProjects	Sucht Projekte nach Kurzname, Name oder Code und liefert nur Projekte, die der Benutzer verwenden darf (optional nur buchbare).	ptm_showProjects.act, ptm_chooseProjectDialog.act, ptm_editTimeEfforts.act

Das einzige schreibende Tool ist doBooking. Alle anderen Tools lesen nur.

Berechtigungsprüfung der Tools

1. **Aktionsberechtigung (Angebot):** Jedes Tool nennt eine oder mehrere erforderliche Aktionen. Dem Sprachmodell wird ein Tool nur angeboten, wenn der Benutzer für mindestens eine davon berechtigt ist; Berechtigungen auf abgeleitete Konfigurationen der Aktion zählen mit. Verweist ein Tool auf eine Aktion, die es im System nicht gibt, wird diese übersprungen (Warnung im Log); gibt es keine einzige bekannte Aktion, ist das Tool gesperrt. Tools von Modulen, deren Aktionen nicht vorhanden oder nicht berechtigt sind, stehen damit nicht zur Verfügung.
2. **Einschränkung durch den Agenten:** Danach wird die Liste auf die "Erlaubten Tools" des Agenten reduziert (nur wenn ein Agent ausgewählt und die Liste nicht leer ist).
3. **Zielprüfung (Ausführung):** Beim Aufruf prüft Webdesk alle Argumente, deren Name auf uid oder uids endet (z.B. personUid, groupUid), auf Berechtigung für das betreffende Objekt. Für jede genannte UID muss der Benutzer für mindestens eine der erforderlichen Aktionen für genau dieses Objekt berechtigt sein (Berechtigung auf Objektebene bzw. passende Konfiguration der Aktion für diese Person). Andernfalls erhält das Modell die Fehlermeldung "Tool not permitted for requested target.", und das Tool wird nicht ausgeführt.

Die Tools sind damit an die Berechtigungen des Benutzers gebunden. Fehler und Ablehnungen sind Teil der Tool-Anzeige im Chat (Arguments / Result).

Schnittstellen (REST)

Die Chat-Oberfläche kommuniziert mit Endpunkten unter /rest/ai. Sie arbeiten immer im Namen des angemeldeten Benutzers.

Endpunkt	Zweck	Hinweise
POST /rest/ai/chat	Sendet eine Nachricht (optional mit Agent, Konversation und Seitenkontext) und liefert Antwort, Tool-Aufrufe und Oberflächen-Effekte.	Leere Nachrichten werden abgewiesen.
GET /rest/ai/agents	Agenten des Mandanten und öffentliche Agenten, alphabetisch.	Befüllt die Agent-Auswahl.
GET /rest/ai/mentions	Personen des Mandanten für die Erwähnung per @Benutzername.	Nur Benutzername, Name und UID.
GET /rest/ai/conversations	Eigene Konversationen (Parameter offset, limit; Standard 50, höchstens 200).	Nur eigene Konversationen.
GET /rest/ai/conversations/{id}/messages	Nachrichten einer eigenen Konversation (Standard 200, höchstens 1000).	HTTP 404, wenn die Konversation nicht dem Benutzer gehört.
POST /rest/ai/conversations/{id}/title	Benennt eine eigene Konversation um (höchstens 255 Zeichen).	
GET /rest/ai/mcp/tools	Liste der Tools (Name, Beschreibung, Parameter), die der angemeldete Benutzer nach seinen Aktionsberechtigungen verwenden darf.	Agenten-Einschränkungen gelten hier nicht.
POST /rest/ai/mcp/tools/{name}	Führt ein Tool direkt aus. Body: {"args": {...}} oder die Argumente unmittelbar als JSON-Objekt. Antwort: {"tool": ..., "result": ...}.	HTTP 404 (unbekanntes Tool), 403 (Aktions- oder Zielberechtigung fehlt), sonst 400.

Die mcp-Endpunkte sind eine einfache MCP-ähnliche Schnittstelle und nutzen dieselbe Berechtigungsprüfung wie der Chat, jedoch ohne Agenten und ohne Sprachmodell: Auch doBooking kann dort aufgerufen werden. Für interne Aufrufe aus der Oberfläche werden die REST-Zugriffsrechte der Aktion ai_assistant.act (/rest/ai/*, GET und POST) herangezogen; externe Aufrufer (z.B. mit API-Schlüssel) benötigen rollenbasierte REST-Zugriffsrechte.

Gesprächsverlauf

- Jede Konversation gehört genau einem Benutzer und seinem Mandanten; andere Benutzer können sie über die Schnittstellen nicht abrufen. Es gibt keine Administrationsmaske für Konversationen.
- Gespeichert werden Konversation (Titel, Besitzer, Mandant, gewählter Agent und Modell, Zeitstempel) und je Nachricht Rolle (Benutzer, Assistent, Tool-Ergebnis), Text, Tool-Name, Tool-Argumente, Tool-Ergebnis (JSON), Fehlertext, Agent, Modell und Zeitstempel (Tabelle AiConversationMessage). Tool-Ergebnisse enthalten damit auch die abgefragten Daten, z.B. Namen und E-Mail-Adressen von Kollegen.
- Der Titel wird aus den ersten 80 Zeichen der ersten Frage gebildet und kann umbenannt werden.
- Der Chat ruft für jede Nachricht das Sprachmodell mit dem bisherigen Verlauf auf (letzte 30 Nachrichten; ältere Teile werden bei Überschreiten eines geschätzten Umfangs von rund 3500 Token

entfernt). Wird eine bestehende Konversation mit dem Kennzeichen "fresh" neu gestartet (Menü "Regenerate fresh" im Chat), werden ihre gespeicherten Nachrichten gelöscht.

- Im Code gibt es keine automatische Löschung oder Aufbewahrungsfrist und keine Löschfunktion für ganze Konversationen. Eine Bereinigung muss gegebenenfalls in der Datenbank erfolgen.
- Fehlermeldungen des Chats (siehe *Fehlersuche*) werden ebenfalls als Assistenten-Nachricht im Verlauf gespeichert.

Datenschutz: Was verlässt das System?

Bei jeder Chat-Anfrage sendet Webdesk an den konfigurierten Endpunkt (Basis-URL des Modells bzw. der globalen Vorgabe):

- **Standard-Prompt** mit dem aktuellen Zeitpunkt, dem Benutzernamen und Regeln für die Verwendung der Tools, ergänzt um den **System Prompt** des Agenten.
- **Angaben zum angemeldeten Benutzer:** UID, Benutzername, Vor- und Nachname, Mitarbeiter-ID, Gültigkeit (von/bis), Download-Adresse des Personenfotos, Sprache, TA-ID, Zeitzone, E-Mail-Adresse, Aktiv-Kennzeichen und Kurzname der Organisationseinheit.
- **Seitenkontext** der aktuell geöffneten Seite: Aktionsname, Menüpfad, Seitentitel, vollständige Seiten-URL, Typ der Aktion, Prozessdefinition sowie die Objekt-ID und URL-Parameter der Seite. Der Benutzer kann diesen Kontext im Panel über das Auge-Symbol ("Show shared context") einblenden.
- die **Nachrichten des Benutzers** samt Gesprächsverlauf (siehe oben) und die **Ergebnisse aller aufgerufenen Tools** (z.B. Personendaten von Kollegen, Buchungen, Abwesenheiten, Urlaubstage, Projektzeiten) sowie die **Beschreibungen und Parameter der angebotenen Tools**.

Der API-Schlüssel dient ausschließlich als Zugangsdaten für den konfigurierten Endpunkt. Die Verarbeitung beim Anbieter richtet sich nach dessen Bedingungen und liegt außerhalb von Webdesk; diese Seite trifft dazu keine Aussage.

Im Server-Log (Logger der Klasse at.workflow.webdesk.ai.impl.AiChatServiceImpl) stehen auf Level INFO der Benutzername, der gewählte Anbieter mit Modell, Basis-URL und Agentenname sowie die Tool-Aufrufe mit Argumenten und Ergebnissen (jeweils auf 500 Zeichen gekürzt). Der API-Schlüssel wird nicht protokolliert.

Einschränkungsmöglichkeiten

Maßnahme	Wirkung
Aktion ai_assistant.act nicht vergeben	Der Benutzer sieht das Symbol nicht und erhält keinen Zugriff auf die Chat-Schnittstelle. Wirksamste Steuerung nach Benutzergruppen.
Lizenz ai nicht erwerben bzw. nicht einspielen	Das Modul steht nicht zur Verfügung.
Keinen API-Schlüssel hinterlegen (weder im Modell noch als Umgebungsvariable, JVM-Parameter oder Systemparameter)	Es erfolgt kein Aufruf an einen externen Anbieter; der Chat antwortet nur mit der Fehlermeldung.
Aktionsberechtigungen der Tools entziehen (z.B. ta_doBooking.act, po_showPersons.act)	Die zugehörigen Tools werden dem Modell nicht angeboten und sind auch über die Schnittstelle gesperrt. Dies ist die verbindliche Einschränkung.
Agenten mit Tool-Auswahl verwenden	Reduziert die angebotenen Tools und damit die übertragenen Daten für Benutzer, die den Agenten verwenden. Greift nicht, wenn der Benutzer "Default agent (auto)" wählt.
Mandantenbindung von Modellen und Agenten	Begrenzt, welche Mandanten welchen Anbieter bzw. Zugang nutzen.

Datenbereinigung	Gespeicherte Konversationen bleiben bestehen, bis sie in der Datenbank entfernt werden.

Fehlersuche

Symptom	Ursache und Prüfung
Kein Assistent-Symbol in der Kopfzeile	Berechtigung für ai_assistant.act fehlt oder Lizenz ai fehlt bzw. das Modul wurde nicht geladen.
"AI chat fallback is active; no API key configured."	In keiner Stufe der Reihenfolge ist ein API-Schlüssel gesetzt. Beachten, dass Umgebungsvariable und JVM-Parameter vor dem Systemparameter gelten und dass im Modus "Default agent (auto)" das Modellfeld nicht berücksichtigt wird.
"AI chat fallback is active; error during model invocation."	Der Aufruf des Anbieters ist fehlgeschlagen (falscher Schlüssel, Modellname, Basis-URL oder Deployment, Netzwerk, bei Azure OpenAI fehlende Basis-URL). Die Ursache steht im Server-Log ("AiChatService error"); die Zeile "AI model resolved" zeigt Anbieter, Modell, Basis-URL und Agent.
"Model returned no text and no tool calls..."	Das Modell bzw. der Anbieter unterstützt keine Tool-Aufrufe für diese Anfrage. Anderes Modell wählen oder im Agenten die Tools reduzieren.
Ein erwartetes Tool wird nicht verwendet	Aktionsberechtigung für das Tool fehlt, das Tool ist beim gewählten Agenten nicht in den "Erlaubten Tools", oder die Zielprüfung lehnt die UID ab ("Tool not permitted for requested target.").
Routing-Praeferenz ohne Wirkung	Die Basis-URL enthält nicht cortecs.ai; der Wert wird ignoriert.

1. /daisy/personalwolke-admin/11418-dsy.html?language=4
2. /daisy/personalwolke-admin/5122-dsy.html?language=4
3. /daisy/personalwolke-admin/5524-dsy.html?language=4