Prüfausgabe — die redaktionelle Gesamtabnahme ist noch nicht abgeschlossen.

Aionda®

Deutsche Ausgabe · 23. November 2026

Feldwerk Produkthandbuch

Datensätze, Datenmodellierung, Automatisierung und Zusammenarbeit

Inhalt

1. Erste Schritte mit Feldwerk
2. Glossar: wichtige Begriffe in Feldwerk
3. Datensätze und Listenansichten
4. Auf einer Datensatzseite arbeiten
5. Globale Suche
6. Anhänge und Themen
7. Objekte und Felder
8. Nachschlagefelder und verwandte Listen
9. Formelfelder
10. Aggregat- und Autonummernfelder
11. Validierungsregeln
12. Formular-Builder
13. Öffentliche Webformulare
14. Datensatzseiten-Builder und Aktionen
15. Berichte
16. Dashboards
17. Mit Flows automatisieren
18. Bildschirm-Flows
19. E-Mail-Vorlagen, Signaturen und Versand
20. Dokumente und Vorlagen
21. Benutzer, Berechtigungen und Freigaben
22. Apps und Erscheinungsbild
23. KI-Assistent und Gedächtnis
24. API, MCP und Integrationen
25. Fehlerbehebung
26. Formelreferenz: 94 Funktionen

Einstieg · Kapitel 1

Erste Schritte mit Feldwerk

Melden Sie sich an, finden Sie Ihren Arbeitsbereich und lernen Sie Ihren ersten Datensatz kennen.

Bevor Sie beginnen

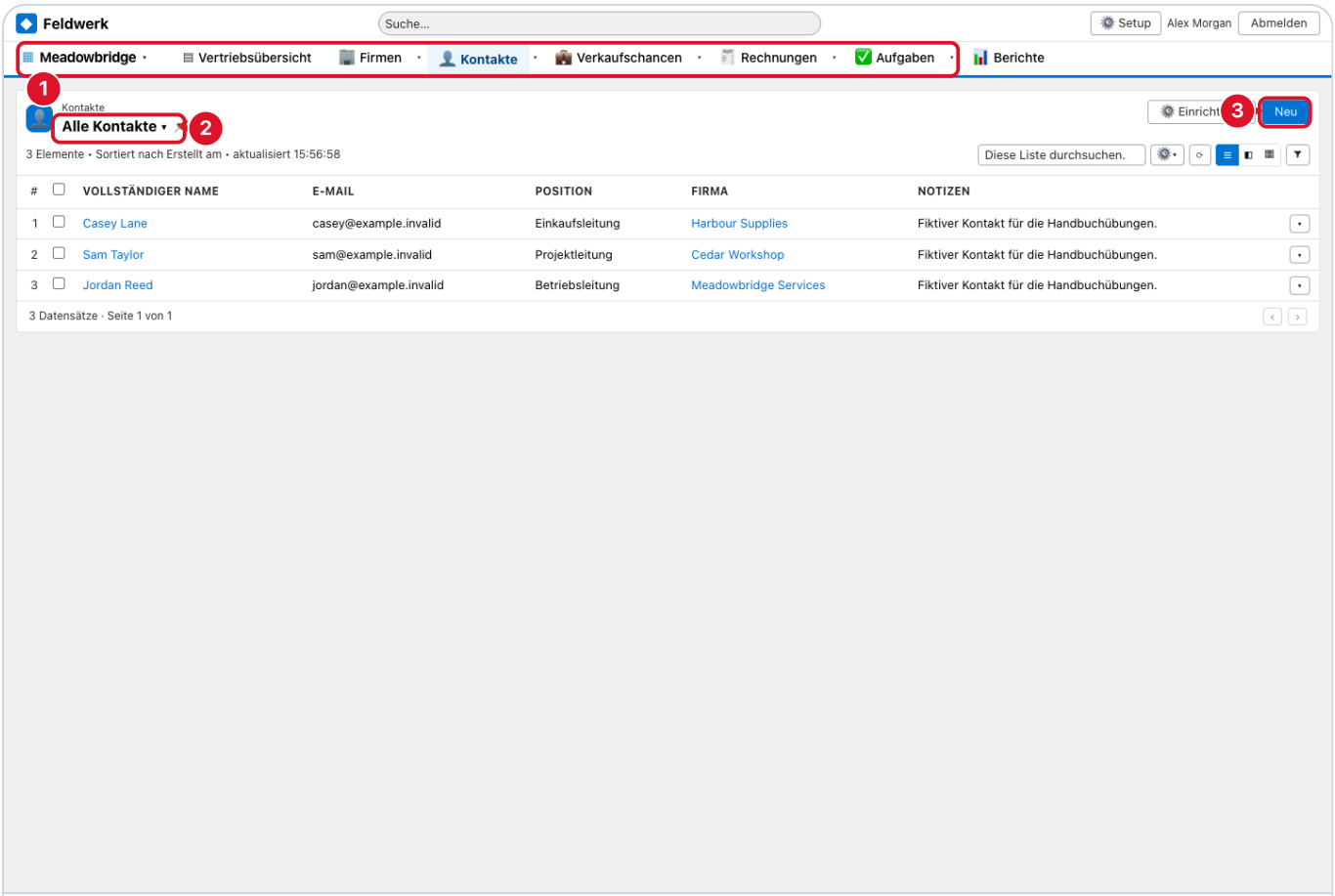
Feldwerk vereint die Datensätze, Formulare, Berichte und Arbeitsabläufe Ihrer Organisation in einem Arbeitsbereich. Ihre Administration bestimmt, welche Objekte und Apps verfügbar sind. Die Beispielobjekte dieses Handbuchs, etwa Kontakte und Aufgaben, zeigen eine mögliche Konfiguration; Ihr Arbeitsbereich kann andere Namen verwenden.

Fragen Sie Ihre Administration nach der Feldwerk-Adresse und Ihren Zugangsdaten. Öffnen Sie die Anmeldeseite unter der Adresse Ihrer Organisation. Sie benötigen ein Konto mit Leseberechtigung für die Datensätze, mit denen Sie arbeiten möchten.

Anmelden

1. Geben Sie Ihre **E-Mail-Adresse** und Ihr **Passwort** ein.
2. Wählen Sie **Anmelden**.
3. Warten Sie, bis sich Ihr Arbeitsbereich öffnet. Erscheint eine Meldung, prüfen Sie Adresse und Zugangsdaten und versuchen Sie es erneut. Falls die Anmeldung weiterhin scheitert, wenden Sie sich mit der Fehlermeldung an Ihre Administration. Geben Sie dabei niemals Ihr Passwort weiter.

Wenn Sie **Ersteinrichtung** und **Konto erstellen** sehen, besitzt die Instanz noch kein Benutzerkonto. Das erste Konto erhält Administratorrechte. Diese Seite ist für die Person bestimmt, die die Instanz einrichtet; sie ist keine allgemeine Registrierungsseite für weitere Benutzer.



1. App oder Objektregister wählen.
2. Listenansicht wählen.
3. Mit Neu einen Kontakt anlegen.

Ihre Arbeit finden

1. Öffnen Sie die App-Auswahl neben dem aktuellen App-Namen, etwa **Meadowbridge**, und wählen Sie den Arbeitsbereich für Ihre Aufgabe.
2. Wählen Sie ein Objektregister. Ein Objekt fasst gleichartige Datensätze zusammen: Kontakte enthält beispielsweise einzelne Kontaktdatensätze.
3. Wählen Sie eine Listenansicht, um die angezeigten Datensätze einzugrenzen. Eine Ansicht verändert die Darstellung; sie erstellt keine separate Kopie der Datensätze.
4. Wählen Sie einen Datensatz, um seine Details zu öffnen. Verwandte Datensätze, Dateien und Aktionen hängen von der konfigurierten Seite und Ihren Berechtigungen ab.

Über die **Suche** finden Sie Datensätze auch über mehrere zugängliche Objekte hinweg. Fehlt ein Datensatz, prüfen Sie zunächst die gewählte Ansicht und ihre Filter. Bitten Sie anschließend gegebenenfalls Ihre Administration, Ihren Zugriff zu prüfen.

Ihre erste Änderung

1. Öffnen Sie einen Testdatensatz, den Sie bearbeiten dürfen.
2. Wählen Sie die Bearbeitungsaktion, ändern Sie ein normales Textfeld und speichern Sie.
3. Prüfen Sie den gespeicherten Wert auf der Datensatzseite. Lehnt eine Validierung die Änderung ab, lesen Sie die Meldung und korrigieren Sie den angegebenen Wert, bevor Sie erneut speichern.

Formel- und Aggregatfelder werden aus anderen Daten berechnet. Sie können ihren Wert nicht durch die Eingabe eines Ersatzwerts ändern. Beim Speichern können außerdem konfigurierte Automatisierungen ausgeführt werden. Verwenden Sie zum Lernen deshalb einen eigenen Testdatensatz.

Wann Sie das Setup benötigen

Mit einem vorhandenen Kontakt oder einer Aufgabe zu arbeiten ist etwas anderes, als deren Struktur zu ändern. Administratoren definieren im **Setup** Objekte, Felder, Formulare, Berechtigungen und Automatisierungen. Wenn Sie ein neues Feld benötigen oder ein erforderliches Objekt nicht sehen, fragen Sie Ihre Administration. Ein fehlendes Register bedeutet für sich genommen nicht, dass die zugehörigen Datensätze gelöscht wurden.

Begriffe kennenlernen

Schlagen Sie unbekannte Begriffe im Glossar nach. Beginnen Sie mit Objekt, Datensatz, Feld und Listenansicht; diese Begriffe beschreiben die meisten Bestandteile des täglichen Arbeitsbereichs.

Glossar: wichtige Begriffe in Feldwerk

Verständliche Erklärungen zu Objekten, Datensätzen, Ansichten, Automatisierung und Zugriff.

Diese Begriffe beschreiben den konfigurierbaren Arbeitsbereich von Feldwerk. Die beispielhaften Objektnamen dienen der Veranschaulichung; Ihre Organisation kann andere Namen wählen.

Objekt

Eine Informationsart, die Ihre Organisation speichert, etwa ein Kontakt oder eine Rechnung. Objekte definieren die Felder, die auf ihren Datensätzen verfügbar sind.

Schritte und Beispiele finden Sie unter Objekte und Felder.

Datensatz

Ein einzelner Eintrag innerhalb eines Objekts, etwa ein Kontakt. Seine Datensatzseite zeigt seine Werte und die für diese Seite konfigurierten verwandten Informationen.

Schritte und Beispiele finden Sie unter Auf einer Datensatzseite arbeiten.

Feld

Ein benannter Wert an einem Datensatz, etwa E-Mail-Adresse, Fälligkeitsdatum oder Status. Der Feldtyp bestimmt, was Sie eingeben können. Pflichtfelder müssen vor dem Speichern einen Wert enthalten.

Schritte und Beispiele finden Sie unter Objekte und Felder.

API-Name

Der stabile technische Name eines Objekts oder Felds. Eine Beschriftung ist der für Menschen angezeigte Name; Integrationen und Konfigurationen verwenden den API-Namen. Das Ändern einer Beschriftung benennt den API-Namen nicht um.

Schritte und Beispiele finden Sie unter Objekte und Felder.

Listenansicht

Eine gespeicherte Darstellung von Datensätzen mit Spalten, Sortierung und Filtern. Mehrere Ansichten können dieselben zugrunde liegenden Datensätze zeigen. Eine Kanban-Ansicht gruppiert Karten anhand eines Auswahllistenfelds in Spalten.

Schritte und Beispiele finden Sie unter Datensätze und Listenansichten.

Nachschlagefeld

Ein Feld, das einen Datensatz mit einem anderen verknüpft. Beispielsweise kann ein Kontakt auf eine Firma verweisen. Wählen Sie den gewünschten Datensatz aus, statt sich nur auf den Namen zu verlassen, wenn mehrere Datensätze gleich heißen.

Schritte und Beispiele finden Sie unter Nachschlagefelder und verwandte Listen.

Verwandte Liste

Datensätze, die über eine Beziehung mit dem aktuellen Datensatz verknüpft sind. Eine Firmenseite kann beispielsweise Kontakte zeigen, die auf diese Firma verweisen. Welche Listen erscheinen, hängt von der Seitenkonfiguration und den Zugriffsrechten ab.

Schritte und Beispiele finden Sie unter Nachschlagefelder und verwandte Listen.

Formular

Eine Anordnung von Feldern und Abschnitten zur Eingabe oder Bearbeitung von Informationen. Über ein öffentliches Webformular können Personen ohne Anmeldung Daten übermitteln; verfügbare Felder und Prüfungen unterscheiden sich von einem internen Formular.

Schritte und Beispiele finden Sie unter Formular-Builder.

Datensatzseite

Die Seite für einen Datensatz. Sie kann Details, verwandte Listen, eine Zeitleiste, Anhänge, Themen und Aktionen kombinieren. Administratoren bestimmen ihr Layout.

Schritte und Beispiele finden Sie unter Auf einer Datensatzseite arbeiten.

Formelfeld

Ein schreibgeschützter Wert, der beim Lesen von Datensätzen aus einem Ausdruck berechnet wird. Ändern Sie seine Eingangswerte oder seine Konfiguration, um das Ergebnis zu ändern.

Schritte und Beispiele finden Sie unter Formelfelder.

Aggregatfeld

Eine schreibgeschützte Zusammenfassung verwandter untergeordneter Datensätze, etwa deren Anzahl oder eine Summe. Die Administration wählt Beziehung, Berechnung und gegebenenfalls einen Filter.

Schritte und Beispiele finden Sie unter Aggregat- und Autonummernfelder.

Autonummernfeld

Eine Nummer, die das System beim Anlegen eines Datensatzes nach einem konfigurierten Format vergibt. Sie wird nicht von Ihnen in das Formular eingegeben.

Schritte und Beispiele finden Sie unter Aggregat- und Autonummernfelder.

Validierungsregel

Eine Regel, die einen Speichervorgang ablehnen und erklären kann, was korrigiert werden muss. Administratoren definieren Bedingung und Meldung; Benutzer korrigieren den Datensatz vor dem nächsten Versuch.

Schritte und Beispiele finden Sie unter Validierungsregeln.

Flow

Eine konfigurierte Automatisierung aus Schritten und Entscheidungen. Je nach Auslöser reagiert sie auf einen gespeicherten Datensatz oder hochgeladenen Anhang, läuft nach Zeitplan oder führt Benutzer durch Bildschirme. Welche Schritte erlaubt sind, hängt vom Auslöser ab.

Schritte und Beispiele finden Sie unter Flows und Automatisierung.

Aktion

Ein Befehl auf einer Datensatzseite oder in einer Liste, etwa einen Datensatz bearbeiten, einen Bildschirm-Flow starten oder ein Dokument erzeugen. Verfügbare Aktionen hängen von Konfiguration und Berechtigungen ab.

Schritte und Beispiele finden Sie unter Datensatzseiten und Aktionen.

Bericht

Eine Auswertung von Datensätzen anhand gewählter Felder, Filter, Gruppierungen und Berechnungen. Ein Dashboard kombiniert konfigurierte Widgets, etwa Berichte und Datensatzlisten, auf einer Seite.

Schritte und Beispiele finden Sie unter Berichte.

App

Ein konfigurierter Arbeitsbereich mit Navigationsregistern. Beim Wechsel der App ändert sich Ihr Navigationskontext; es entsteht kein weiteres Benutzerkonto.

Schritte und Beispiele finden Sie unter Apps und Erscheinungsbild.

Berechtigungen und Freigaben

Berechtigungen bestimmen, welche Operationen und Felder Sie verwenden dürfen. Freigaben bestimmen den Zugriff auf einzelne Datensätze. Einen Datensatz sehen zu können erlaubt nicht automatisch dessen Bearbeitung oder Löschung.

Schritte und Beispiele finden Sie unter Benutzer, Berechtigungen und Freigaben.

Anhang

Eine an einem Datensatz gespeicherte Datei. Der Zugriff wird über den Datensatz geprüft. Das Anhängen einer PDF-Datei macht deren gesamten Text nicht automatisch in der globalen Datensatzsuche verfügbar.

Schritte und Beispiele finden Sie unter Anhänge und Themen.

Thema

Ein Schlagwort an einem Datensatz, das ihn beschreibt oder einordnet. Themen ergänzen Ihre Felder und ersetzen nicht die Struktur des Objekts.

Schritte und Beispiele finden Sie unter Anhänge und Themen.

API-Token

Ein Zugangsschlüssel, mit dem eine Integration im Namen seines Eigentümers handelt. Halten Sie ihn geheim und widerrufen Sie ihn, sobald er nicht mehr benötigt wird. Sein Zugriff richtet sich nach den Berechtigungen des Eigentümers.

Schritte und Beispiele finden Sie unter API-Tokens und MCP.

MCP

Model Context Protocol: ein Verfahren, mit dem ein kompatibler Assistent Feldwerk-Werkzeuge entdecken und aufrufen kann. Diese Werkzeuge verwenden dieselben berechtigungsgeprüften Anwendungsoperationen wie andere unterstützte Zugangswege.

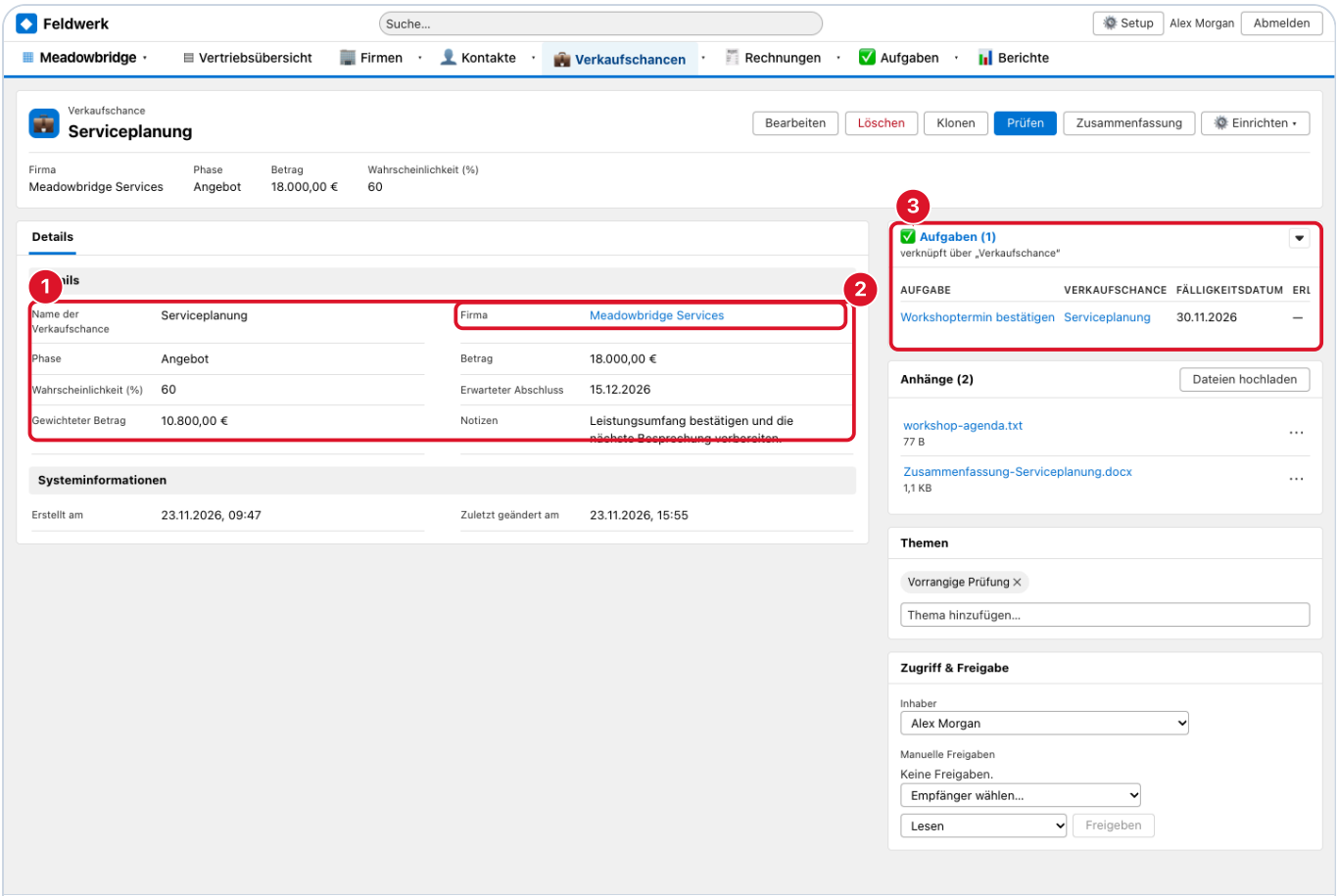
Schritte und Beispiele finden Sie unter API-Tokens und MCP.

Dashboard

Eine Seite, die Berichte, Listen, Bildschirm-Flows und Text ohne einzelnen aktuellen Datensatz kombiniert. Fügen Sie sie als Register zu einer App hinzu. Siehe Dashboards.

Mit der Arbeit beginnen

Folgen Sie Erste Schritte, um sich anzumelden und Ihren ersten Datensatz zu finden.



1. Felder enthalten die Werte dieses Datensatzes.
2. Das Nachschlagefeld Firma verknüpft einen anderen Datensatz.
3. Eine verwandte Liste zeigt mit dieser Verkaufschance verknüpfte Datensätze.

Datensätze und Listenansichten

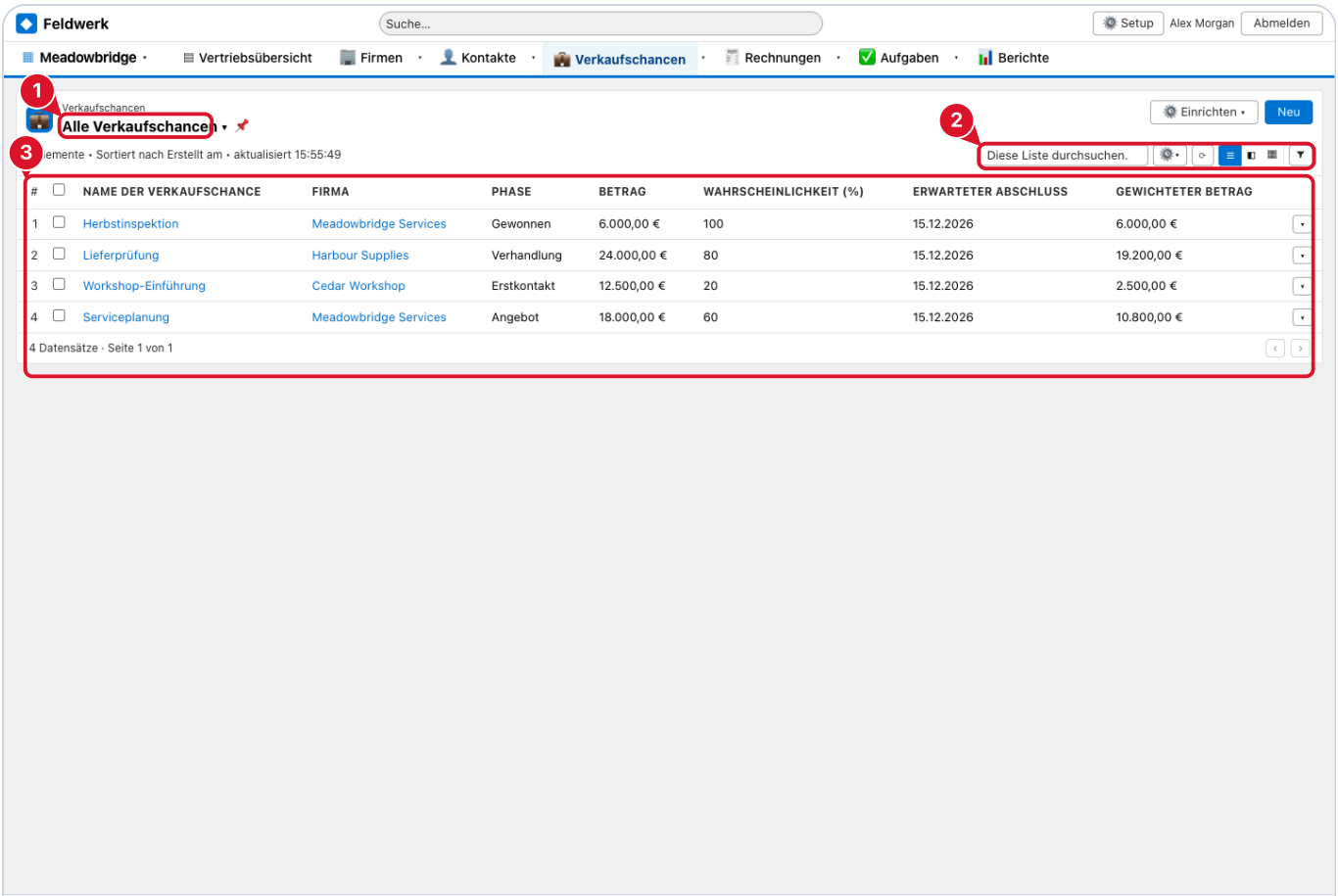
Datensätze in Tabellen-, geteilter und Kanban-Ansicht finden, ordnen und aktualisieren.

Objekt und Ansicht wählen

Öffnen Sie ein Objektregister, etwa **Verkaufschancen**. Der Ansichtsname unter dem Objektnamen bezeichnet die gespeicherte Listenkonfiguration. Über den Pfeil wählen Sie eine andere Ansicht oder **Neue Listenansicht**. Eine Listenansicht speichert Darstellungsoptionen wie Felder, Filter, Sortierung und Anzeigemodus; die zugrunde liegenden Datensätze bleiben dieselben.

Öffnen Sie zum Anlegen einer Ansicht die Ansichtsauswahl, wählen Sie **Neue Listenansicht**, geben Sie einen Namen ein und speichern Sie. Die neue Ansicht übernimmt zunächst die Spalten der aktuellen Ansicht; Filter und weitere Einstellungen konfigurieren Sie separat. Zum Umbenennen öffnen Sie **Listenansicht-Steuerung → Umbenennen**, ändern den Namen und speichern. Die Ansicht hat eine eigene URL, die Sie als Lesezeichen speichern oder mit zugriffsberechtigten Personen teilen können.

Mit der Stecknadel neben dem Ansichtsnamen legen Sie die Standardlistenansicht fest. Eine Ansicht ist keine private Kopie der Daten. Wenn Sie einen Datensatz in einer Ansicht ändern, gilt diese Änderung überall, wo er angezeigt wird.



1. Listenansicht wählen oder anheften.
2. Suchen, konfigurieren oder Darstellungsart ändern.
3. Datensätze öffnen oder ihre Zeilenaktionen verwenden.

Anlegen, bearbeiten und klonen

1. Wählen Sie in der Objektliste **Neu**.
2. Füllen Sie die mit einem Stern gekennzeichneten Pflichtfelder aus. Wählen Sie bei einem Nachschlagefeld den gewünschten verknüpften Datensatz aus den Vorschlägen.
3. Wählen Sie **Speichern** und prüfen Sie das Ergebnis. Verhindert eine Validierung das Speichern, korrigieren Sie die angegebenen Werte. Das Schließen des Dialogs speichert die Eingaben nicht.

Über das Aktionsmenü einer Zeile können Sie einen bestehenden Datensatz anzeigen oder bearbeiten. Auf der Datensatzseite stehen die konfigurierten Aktionen zur Verfügung, gegebenenfalls auch **Klonen**. Ein Klon ist ein neuer Datensatz: Prüfen Sie kopierte Werte,

Pflichtfelder und eindeutige Werte vor dem Speichern. Berechnete Felder und automatische Nummern richten sich nach ihren Felddefinitionen. Ein Klon ist keine zweite Identität des Originals.

Liste durchsuchen und sortieren

Geben Sie Text in **Diese Liste durchsuchen…** ein, um die aktuelle Objektliste einzugrenzen. Löschen Sie den Suchtext, um wieder die normalen Ergebnisse der Ansicht zu sehen. Im Gegensatz dazu durchsucht das Suchfeld in der globalen Kopfzeile mehrere zugängliche Objekte.

Wählen Sie eine Spaltenüberschrift zum Sortieren oder fokussieren Sie sie mit Tab und drücken Sie Eingabe oder Leertaste. Die Statuszeile der Liste zeigt das aktive Sortierfeld. In der **Listenansicht-Steuerung** können Sie die Sortierung in der Ansicht speichern oder die Spaltensortierung zurücksetzen. Nach Änderungen an anderer Stelle wählen Sie **Aktualisieren**. Bei umfangreichen Ergebnissen verwenden Sie die Seitennavigation am unteren Rand.

Sichtbare Spalten wählen

1. Öffnen Sie **Listenansicht-Steuerung → Anzuzeigende Felder auswählen**.
2. Wählen Sie ein Feld unter Verfügbare Felder oder Sichtbare Felder. Mit Hinzufügen beziehungsweise Entfernen verschieben Sie es zwischen den Listen. Sie können ein Feld mit Tab fokussieren und mit Eingabe oder Leertaste auswählen.
3. Ordnen Sie die sichtbaren Spalten mit Nach oben und Nach unten an.
4. Lassen Sie mindestens eine Spalte sichtbar und speichern Sie.

Das Ausblenden einer Spalte ändert nur die Darstellung. Es entfernt weder das Feld noch schützt es dessen Daten. Der Feldzugriff wird über Berechtigungen geregelt.

Datensätze filtern

1. Wählen Sie **Filter**.
2. Wählen Sie den Eigentümerbereich, beispielsweise alle oder Ihre eigenen Datensätze.
3. Wählen Sie **Filter hinzufügen** und anschließend Feld, Operator und Wert.
4. Ergänzen Sie bei Bedarf weitere Bedingungen und wählen Sie **Anwenden**.

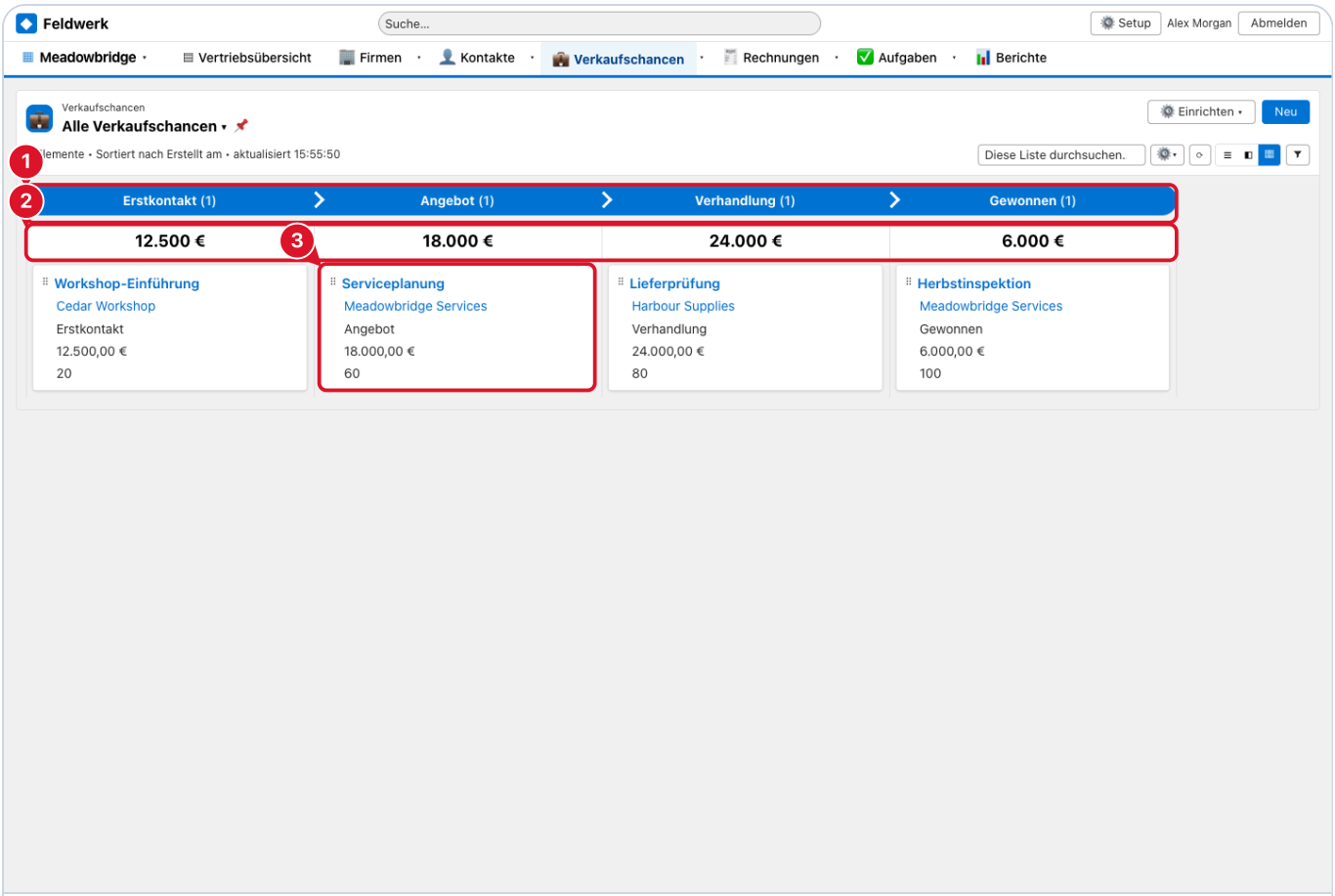
Normalerweise müssen alle Bedingungen zutreffen. Für Alternativen ergänzen Sie eine **Filterlogik** mit nummerierten Bedingungen, etwa `1 AND (2 OR 3)`. Prüfen Sie Vorschau und Ergebnisdatensätze. Ein Link aus einer verwandten Liste kann zusätzlich einen Beziehungsfilter setzen. Mit **Beziehungsfilter entfernen** verlassen Sie diese Einschränkung.

Datumsbedingungen können ein festes Datum, einen relativ zu heute berechneten Wert oder eine eigene Formel wie `TODAY() - 30` verwenden. Relative Werte ändern sich mit dem Datum; ein festes Datum bleibt unverändert. Kann ein Ausdruck nicht ausgewertet werden, trifft die Bedingung nie zu. Korrigieren Sie den Ausdruck, statt daraus auf eine leere Liste zu schließen.

Tabelle, geteilte Ansicht und Kanban

Die **Tabellenansicht** eignet sich zum Vergleich von Spalten. In der **geteilten Ansicht** bleibt die Datensatzliste links sichtbar und der ausgewählte Datensatz wird daneben geöffnet. Wählen Sie einen anderen Datensatz, um ihn zu prüfen, ohne die Liste zu verlassen.

Die **Kanban-Ansicht** gruppiert Karten nach einem Auswahllistenfeld. Unter **Kanban-Einstellungen** wählen Sie das Gruppierungsfeld, optional ein Summenfeld sowie Sichtbarkeit und Reihenfolge der Spalten. Unser Beispiel gruppiert Verkaufschancen nach Phase und summiert den Betrag. Ein Auswahllistenfeld ist erforderlich.



1. Jede Spalte steht für einen Auswahllistenwert.
2. Die Betragssummen fassen die geladenen Karten zusammen.
3. Eine Karte steht für einen Datensatz; beim Verschieben ändert sich seine Phase.

Wenn Sie eine Karte in eine andere Spalte ziehen, ändern Sie ihren Auswahllistenwert. Das ist ein echter Speichervorgang; Validierungen, Berechtigungen und Automatisierungen gelten weiterhin. Scheitert das Verschieben, lesen Sie die Meldung und prüfen Sie die tatsächliche Phase des Datensatzes. Beachten Sie bei großen Datenbeständen den Grenzwert-Hinweis des Boards: Sichtbare Karten und Spaltensummen können sich nur auf den geladenen Teil beziehen.

Listenaktionen und Löschen

Administratoren können unter **Aktionsknöpfe …** aktive Bildschirm-Flows hinzufügen. Diese Schaltflächen starten den konfigurierten Flow ohne einzelnen Datensatzkontext. Ihre Wirkung hängt vom Flow ab. Gehen Sie nicht davon aus, dass angekreuzte Zeilen automatisch an jede Aktion übergeben werden.

Das Löschen einzelner Zeilen und das Stapellöschen erfordern eine Bestätigung. Wählen Sie einzelne Zeilenkästchen oder das Kästchen im Tabellenkopf, um die aktuell geladene Seite auszuwählen. Die Löschschaltfläche zeigt die Anzahl der ausgewählten Datensätze. Prüfen Sie vor dem Bestätigen Anzahl und Auswahl. Abbrechen erhält die Datensätze; bestätigen Sie Löschen nur bei korrekter Auswahl. Das Löschen einer Listenansicht entfernt deren Konfiguration, nicht die Datensätze. Das Löschen von Datensätzen ist ein anderer Vorgang und kann sich auf verknüpfte Informationen auswirken.

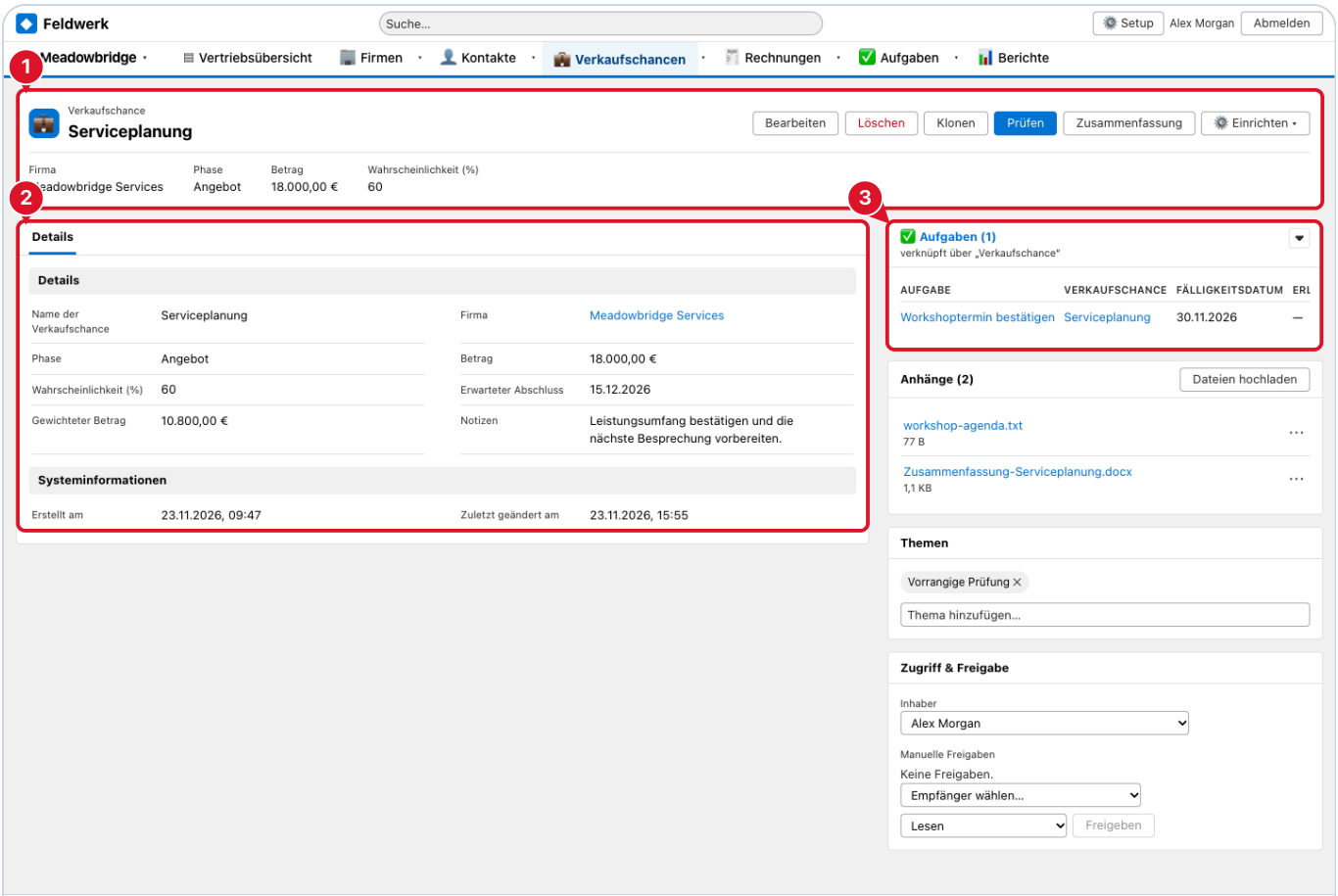
Auf einer Datensatzseite arbeiten

Einen Datensatz lesen, seinen Beziehungen folgen und die für Ihre Arbeit konfigurierten Aktionen nutzen.

Den richtigen Datensatz öffnen

Wählen Sie einen Datensatznamen in einer Liste, einem Suchergebnis oder einem Nachschlagelink. Prüfen Sie Objektname und Datensatztitel vor dem Bearbeiten. Namen sind nicht zwingend eindeutig; unterscheiden Sie ähnlich benannte Datensätze anhand der übrigen Felder.

Eine Datensatzseite kombiniert von Ihrer Administration gewählte Komponenten. Ihre Seite kann andere Register, Felder, verwandte Listen und Schaltflächen als das Handbuchbeispiel enthalten. Die Beispiel-Verkaufschance zeigt eine Zusammenfassung, Details, verwandte Aufgaben, Dateien, Themen und Zugriff & Freigabe.



1. Datensatz prüfen und Aktion wählen.
2. Felder und Systemzeitstempel lesen.
3. Verwandte Aufgaben über die konfigurierte Beziehung öffnen.

Details lesen

Der obere Bereich fasst wichtige Werte zusammen. Der Detailbereich enthält die konfigurierten Felder. Ein blauer Nachschlagewert öffnet den verknüpften Datensatz; er kopiert dessen Informationen nicht in ein separates Objekt.

Formel- und Aggregatwerte werden berechnet. Automatische Nummern vergibt Feldwerk. Schreibgeschützte Felder lassen sich nicht über die normale Eingabe bearbeiten. Die Systeminformationen zeigen Erstellungs- und Änderungszeitpunkte. Damit können Sie prüfen, ob eine aktuelle Änderung am Datensatz angekommen ist.

Eine Änderung vornehmen

1. Wählen Sie **Bearbeiten** oder eine verfügbare Feldbearbeitung auf der Seite.

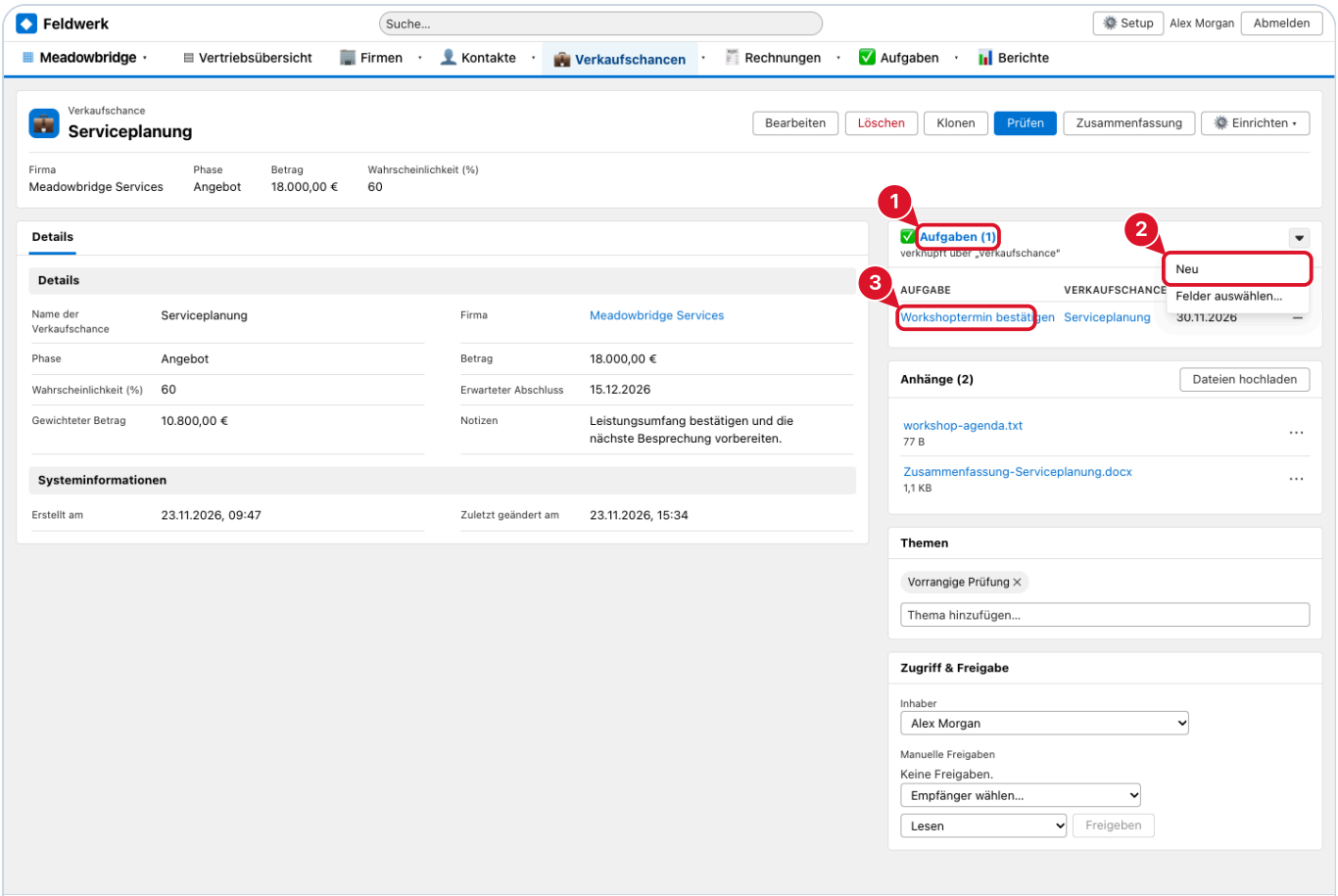
2. Ändern Sie den gewünschten Wert. Suchen und wählen Sie bei einer Beziehung den richtigen Zieldatensatz.
3. Wählen Sie **Speichern**.
4. Prüfen Sie den angezeigten Wert nach dem Speichern. Verhindert eine Meldung die Änderung, korrigieren Sie den Wert und versuchen Sie es erneut.

Mit **Abbrechen** verwerfen Sie ungespeicherte Eingaben. Ein erfolgreicher Speichervorgang kann auch Automatisierungen auslösen; prüfen Sie deshalb abhängige Werte. Bei der Handbuch-Verkaufschance ergeben Betrag `18000` und Wahrscheinlichkeit `60` einen gewichteten Betrag von `10800`.

Mit verwandten Informationen arbeiten

Eine verwandte Liste zeigt Datensätze, die über die unter ihrer Überschrift benannte Beziehung verbunden sind. Die Aufgabenliste der Verkaufschance enthält Aufgaben, deren Nachschlagefeld Verkaufschance auf diesen Datensatz verweist. Wählen Sie einen verwandten Datensatz, um ihn zu öffnen. Bedienelemente wie Alle anzeigen und Zeilenmenüs hängen von der konfigurierten Komponente und Ihrem Zugriff ab.

1. Öffnen Sie in der verwandten Aufgabenliste den Aktionspfeil neben der Listenüberschrift und wählen Sie **Neu**.
2. Geben Sie den Betreff und die weiteren erforderlichen Aufgabendaten ein. Prüfen Sie, dass Verkaufschance bereits auf den aktuellen Datensatz verweist.
3. Speichern Sie und prüfen Sie, dass die Aufgabe in dieser verwandten Liste erscheint.
4. Über das Aktionsmenü der Aufgabenzeile können Sie sie bearbeiten oder löschen. Das Löschen erfordert eine Bestätigung und entfernt die Aufgabe, nicht die übergeordnete Verkaufschance.



1. Alle mit dieser Verkaufschance verknüpften Aufgaben öffnen.
2. Eine Aufgabe anlegen, deren Beziehung zur Verkaufschance bereits ausgefüllt ist.
3. Eine einzelne verwandte Aufgabe öffnen; ihr Zeilenmenü bietet die erlaubten Bearbeitungs- und Löschaktionen.

Wählen Sie die Überschrift **Aufgaben (Anzahl)**, um die vollständige Aufgabenliste mit einem Beziehungsfilter zu öffnen. **Beziehungsfilter entfernen** zeigt die Liste ohne diese Beziehungseinschränkung; die eigenen Filter der gespeicherten Ansicht gelten weiterhin. Anlegen, Bearbeiten und Löschen erfordern die jeweiligen Objektberechtigungen.

Enthält die Seite eine Aktivitätszeitleiste oder einen E-Mail-Bereich, verwenden Sie deren konfigurierte Bedienelemente zum Prüfen oder Verknüpfen von Einträgen. Eine Datensatzseite kann auch Anhänge eines verknüpften Datensatzes anzeigen. Prüfen Sie die Bereichsüberschrift, bevor Sie entscheiden, wo eine Datei hingehört.

Datensatzaktionen verstehen

Zu den Standardaktionen gehören Bearbeiten, Löschen und Klonen, sofern sie konfiguriert sind. Andere Schaltflächen können einen Bildschirm-Flow starten, ein Plugin aufrufen, ein Dokument erzeugen oder eine E-Mail-Aktion öffnen. Ihre Beschriftungen legt die

Administration fest.

Im Beispiel öffnet **Prüfen** einen geführten Bildschirm für den nächsten Schritt. **Zusammenfassung** erzeugt eine DOCX-Datei aus der Verkaufschance. Warten Sie das Ergebnis einer Aktion ab, bevor Sie sie erneut ausführen. Wenn Sie eine unveränderte DOCX-Datei erneut erzeugen, wird sie erneut heruntergeladen; ein identischer vorhandener Anhang wird dabei erkannt.

Dateien, Themen und Zugriff

Unter Dateien laden Sie Anhänge hoch oder öffnen sie. Mit Themen ordnen Sie Datensätze durch Schlagwörter. Zugriff & Freigabe zeigt, sofern vorhanden und erlaubt, den Eigentümer und Bedienelemente für ausdrückliche Freigaben. Eine Layoutänderung erteilt keinen Zugriff. Eine fehlende Aktion kann an Berechtigungen, Konfiguration oder dem aktuellen Datensatzzustand liegen.

Bewusst klonen oder löschen

Klonen öffnet einen neuen Datensatz auf Grundlage der Quellwerte. Prüfen Sie ihn vor dem Speichern. Löschen verlangt eine Bestätigung; kontrollieren Sie darin den Datensatznamen. Das Abbrechen eines Dialogs macht eine bereits abgeschlossene Löschung nicht rückgängig.

Wenn die Seite nicht stimmt

Aktualisieren Sie nach Änderungen an anderer Stelle. Fehlt ein Feld, prüfen Sie die aktive Seiten- und Formularkonfiguration sowie die Feldberechtigungen. Ist eine verwandte Liste leer, prüfen Sie die Beziehung an den untergeordneten Datensätzen und Ihre Sichtbarkeit. Nennen Sie Ihrer Administration bei Rückfragen Objekt, Datensatz und genaue Meldung.

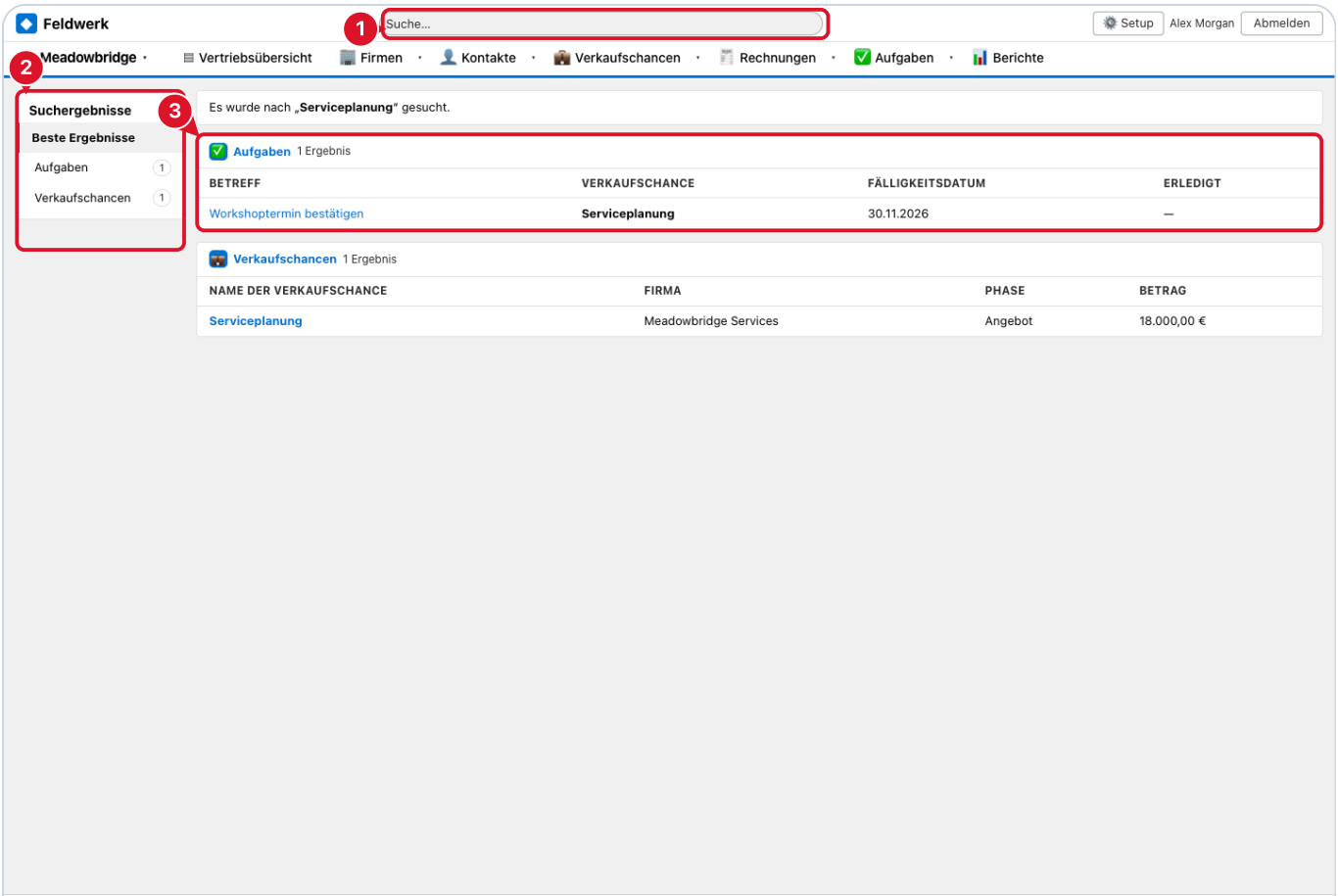
Globale Suche

Datensätze objektübergreifend finden und den Umfang der Ergebnisse verstehen.

Die globale Suche verwenden

1. Wählen Sie **Suchen…** in der Kopfzeile der Anwendung. Auf schmalen Bildschirmen öffnen Sie das Suchfeld über das Suchsymbol.
2. Geben Sie mindestens zwei Zeichen ein. Warten Sie, bis die Vorschläge geladen sind.
3. Wählen Sie einen vorgeschlagenen Datensatz oder drücken Sie **Eingabe**, um die Suchergebnisse zu öffnen.

Beim Fokussieren eines leeren Suchfelds können zuletzt angesehene Datensätze erscheinen. Das sind Navigationshilfen und keine neue Suche. Wenn Sie aus einem Objekt heraus suchen, wird dieses Objekt bei der Reihenfolge der Ergebnisse bevorzugt.



1. Suchbegriff in die globale Kopfzeile eingeben.
2. Beste Ergebnisse oder eine Objektgruppe wählen.
3. Einen passenden Datensatz öffnen und seine Werte prüfen.

Ergebnisse lesen

Ergebnisse werden nach Objekt gruppiert. Beste Ergebnisse zeigt eine objektübergreifende Übersicht; über die Objektauswahl grenzen Sie die angezeigte Gruppe ein. Auf schmalen Bildschirmen steht diese Auswahl oberhalb der Ergebnisse. Mit Mehr anzeigen laden Sie gegebenenfalls weitere Ergebnisse. Wählen Sie einen Datensatznamen, um ihn zu öffnen.

Im Beispiel findet die Suche nach **Serviceplanung** die Verkaufschance und Datensätze, deren gespeicherte Werte denselben Text enthalten, auch in Anzeigewerten von Beziehungen. Der Ergebnistitel muss nicht das übereinstimmende Feld sein. Lesen Sie die Zeile und öffnen Sie den Datensatz, bevor Sie ihn für ein Duplikat halten.

Wenn Sie die öffentliche ID eines Datensatzes kennen, fügen Sie die vollständige ID in die globale Suche ein. Eine exakte ID-Übereinstimmung wird zuerst angezeigt. Das hilft bei ähnlichen Namen; Ihre normalen Objekt- und Datensatzberechtigungen gelten weiterhin.

Globale Suche und Listensuche

Die Suche in der Kopfzeile umfasst die Objekte, auf die Sie zugreifen dürfen. **Diese Liste durchsuchen…** gehört dagegen zu einer Objektliste und arbeitet innerhalb ihres Umfangs. Listenfilter, Eigentümerbereich oder Beziehungsfilter können einen Datensatz ausblenden, obwohl er über einen anderen berechtigten Zugangsweg auffindbar ist. Leeren Sie die Listensuche und prüfen Sie die Filter, wenn ein Datensatz zu fehlen scheint.

Was durchsucht wird

Die aktuelle globale Suche durchsucht gespeicherte Datensatzwerte. Sie ist kein allgemeiner Volltextindex aller angehängten Dateien. Durch das Hochladen einer PDF-Datei wird deren gesamter Text nicht automatisch durchsuchbar.

Soll Dokumenttext durchsuchbar sein, kann eine Administration einen durch Anhänge ausgelösten Flow konfigurieren. Dieser extrahiert unterstützten Text und schreibt ihn über die normale Datensatzverarbeitung in ein geeignetes Feld. Feld, Objekt, Berechtigungen und Extraktionsregeln bestimmen das Suchverhalten. Versprechen Sie keine Treffer in Dateiinhalten, bevor dieser Ablauf eingerichtet und getestet wurde.

Wenn Sie einen Datensatz nicht finden

1. Prüfen Sie die Schreibweise und versuchen Sie einen kürzeren, charakteristischen Teil eines gespeicherten Werts.
2. Verwenden Sie für die globale Suche mindestens zwei Zeichen.
3. Prüfen Sie, ob Sie in der globalen Kopfzeile oder nur in einer gefilterten Liste suchen.
4. Öffnen Sie das erwartete Objekt direkt und prüfen Sie seine Ansichten.
5. Bitten Sie Ihre Administration, Objektzugriff, Feldzugriff und Datensatzsichtbarkeit zu prüfen, wenn der Datensatz weiterhin nicht auffindbar ist.

Keine Treffer beweisen nicht, dass ein Datensatz gelöscht wurde. Die Suche berücksichtigt Zugriffsrechte. Neu extrahierter Dokumenttext muss zunächst in einem durchsuchbaren Datensatzfeld gespeichert sein.

Anhänge und Themen

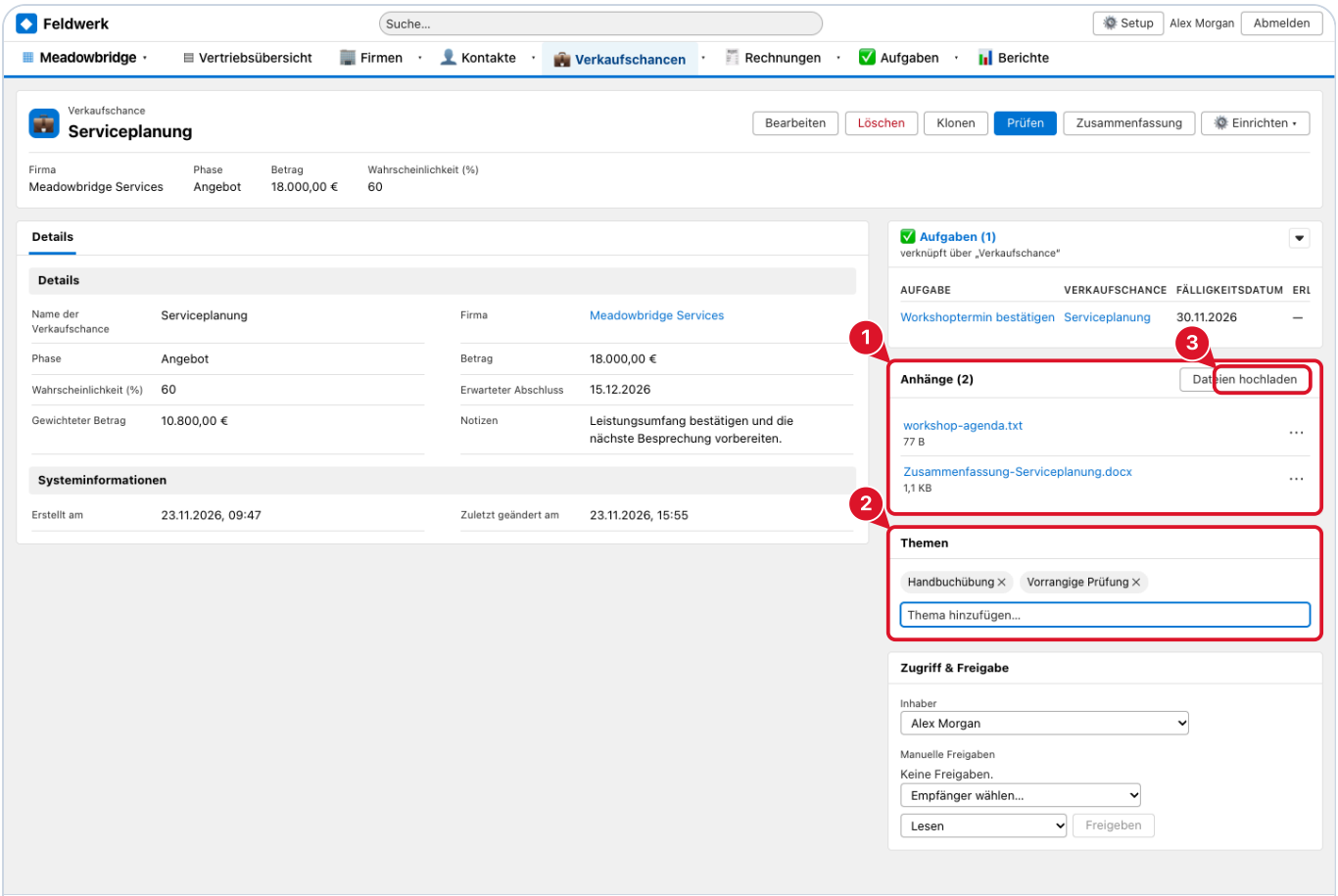
Begleitdateien am Datensatz aufbewahren und zusammengehörige Arbeit mit Themen ordnen.

Eine Datei am richtigen Datensatz anhängen

Öffnen Sie den Datensatz und suchen Sie den Bereich Dateien. Prüfen Sie seine Überschrift: Manche Seitenlayouts zeigen Dateien eines verknüpften Datensatzes. Die sichtbare Seite allein identifiziert daher nicht immer den Eigentümer des Anhangs.

1. Wählen Sie **Dateien hochladen** oder ziehen Sie Dateien in den Uploadbereich. Ist der Bereich leer, können Sie ihn auch mit Tab fokussieren und mit Eingabe oder Leertaste die Dateiauswahl öffnen.
2. Wählen Sie eine oder mehrere Dateien und warten Sie, bis der Upload abgeschlossen ist.
3. Prüfen Sie, dass jeder Dateiname im Bereich erscheint.
4. Meldet die Seite einen fehlgeschlagenen Upload, prüfen Sie vor einem erneuten Versuch Dateiname und Fehler.

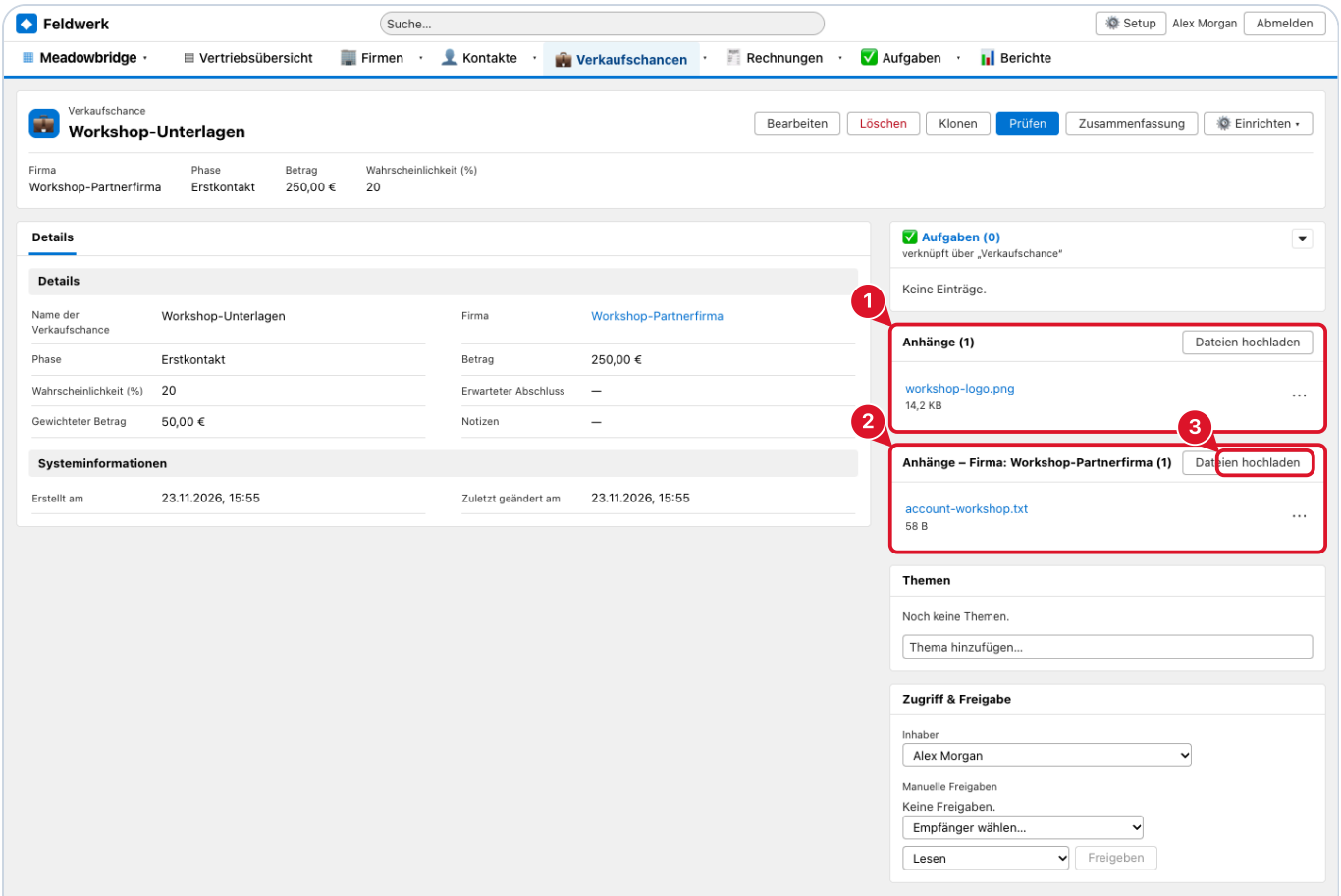
Die aktuelle Uploadgrenze beträgt 25 MB pro Datei. Beim Hochladen mehrerer Dateien können einzelne Uploads erfolgreich sein. Gehen Sie nicht davon aus, dass die gesamte Auswahl fehlgeschlagen ist, nur weil eine Datei abgelehnt wurde. Dateioperationen unterliegen den Zugriffsregeln des Datensatzes.



1. Anhänge dieses Datensatzes prüfen.
2. Themen hinzufügen oder entfernen.
3. Begleitdateien hochladen.

Dateien eines verknüpften Datensatzes

Administratoren können eine Seite so konfigurieren, dass sie Dateien von Datensätzen einbezieht, die über Nachschlagefelder verknüpft sind. Im Beispiel hat die Verkaufschance einen eigenen Dateibereich und einen separaten Bereich **Dateien – Firma: Workshop-Partnerfirma**. Ein Upload über den Firmenbereich speichert die Datei an dieser Firma. Dadurch entsteht keine weitere Kopie an der Verkaufschance. Prüfen Sie vor dem Hochladen oder Löschen die Bereichsüberschrift.



1. Dateien der Verkaufschance.
2. Dateien der verknüpften Firma.
3. Diese Uploadschaltfläche fügt Dateien zur Firma hinzu.

Einen Anhang öffnen oder entfernen

Wählen Sie den Dateinamen, um den Anhang herunterzuladen, und öffnen Sie die heruntergeladene Datei mit einer geeigneten Anwendung. Der Dateinamen-Link im Dateibereich ist eine Downloadaktion und öffnet keine Vorschau innerhalb der Seite. Im Menü **Weitere Aktionen** des Anhangs finden Sie die verfügbaren Aktionen einschließlich Löschen. Prüfen Sie den Dateinamen vor dem Löschen. Das Entfernen eines Anhangs löscht nicht den übergeordneten Datensatz.

Auch erzeugte Dokumente können hier erscheinen. Soweit erlaubt, hängt die Dokumentaktion ihre Ausgabe normalerweise zusätzlich zum Download an. Eine unveränderte erzeugte DOCX-Datei kann den identischen Anhang wiederverwenden; ein geändertes Dokument ist eine andere Ausgabe. Prüfen Sie die Ergebnismeldung der Aktion.

Themen hinzufügen und entfernen

1. Suchen Sie den Bereich **Themen**.
2. Geben Sie Text in **Thema hinzufügen…** ein.
3. Wählen Sie einen vorhandenen Vorschlag oder drücken Sie Eingabe, um das eingegebene Thema hinzuzufügen.
4. Prüfen Sie, dass das Thema als Markierung erscheint.

Über das Entfernen-Symbol einer Markierung lösen Sie das Thema von diesem Datensatz. Damit wird die Zuordnung entfernt, nicht der Datensatz gelöscht. Verwenden Sie möglichst einheitliche vorhandene Themennamen, damit sich zusammengehörige Arbeit verlässlich gruppieren lässt.

Themen ergänzen strukturierte Felder. Nutzen Sie Felder für Werte, die einen definierten Typ, Berechnungen, Validierungen oder geregelte Geschäftszustände benötigen. Ein Thema ersetzt keine Eigentümerregel oder Berechtigung.

Dateien und Suche sind getrennt

Ein Anhang macht seinen Inhalt nicht automatisch zum Bestandteil der globalen Datensatzsuche. Für durchsuchbaren Dokumenttext konfigurieren Sie eine Extraktion in ein Datensatzfeld. Ein durch Anhänge ausgelöster Flow kann dies mit der unterstützten Extraktionsaktion und der normalen Datensatzaktualisierung erledigen.

Häufige Probleme lösen

- **Datei fehlt nach dem Upload:** Warten Sie das Ergebnis ab, prüfen Sie den gemeldeten Dateinamen und aktualisieren Sie die Seite. Prüfen Sie, ob sich der Bereich auf einen anderen, verknüpften Datensatz bezieht.
- **Datei zu groß:** Verkleinern Sie sie unter die unterstützte Grenze und versuchen Sie es erneut.
- **Zugriff verweigert:** Bitten Sie Eigentümer oder Administration, die Datensatz- und Objektberechtigungen zu prüfen.
- **Datei lässt sich nicht öffnen:** Prüfen Sie zunächst, ob der Download vollständig ist. Verwenden Sie anschließend eine Anwendung, die das Format unterstützt.
- **Thema nicht hinzugefügt:** Lesen Sie die Fehlermeldung und prüfen Sie den Bearbeitungszugriff. Bloßes Tippen ohne Bestätigung erstellt noch keine Zuordnung.

Objekte und Felder

Definieren Sie die Datensatzarten Ihrer Organisation und die Werte, die jeder Datensatz speichern kann.

Mit dem Informationsmodell beginnen

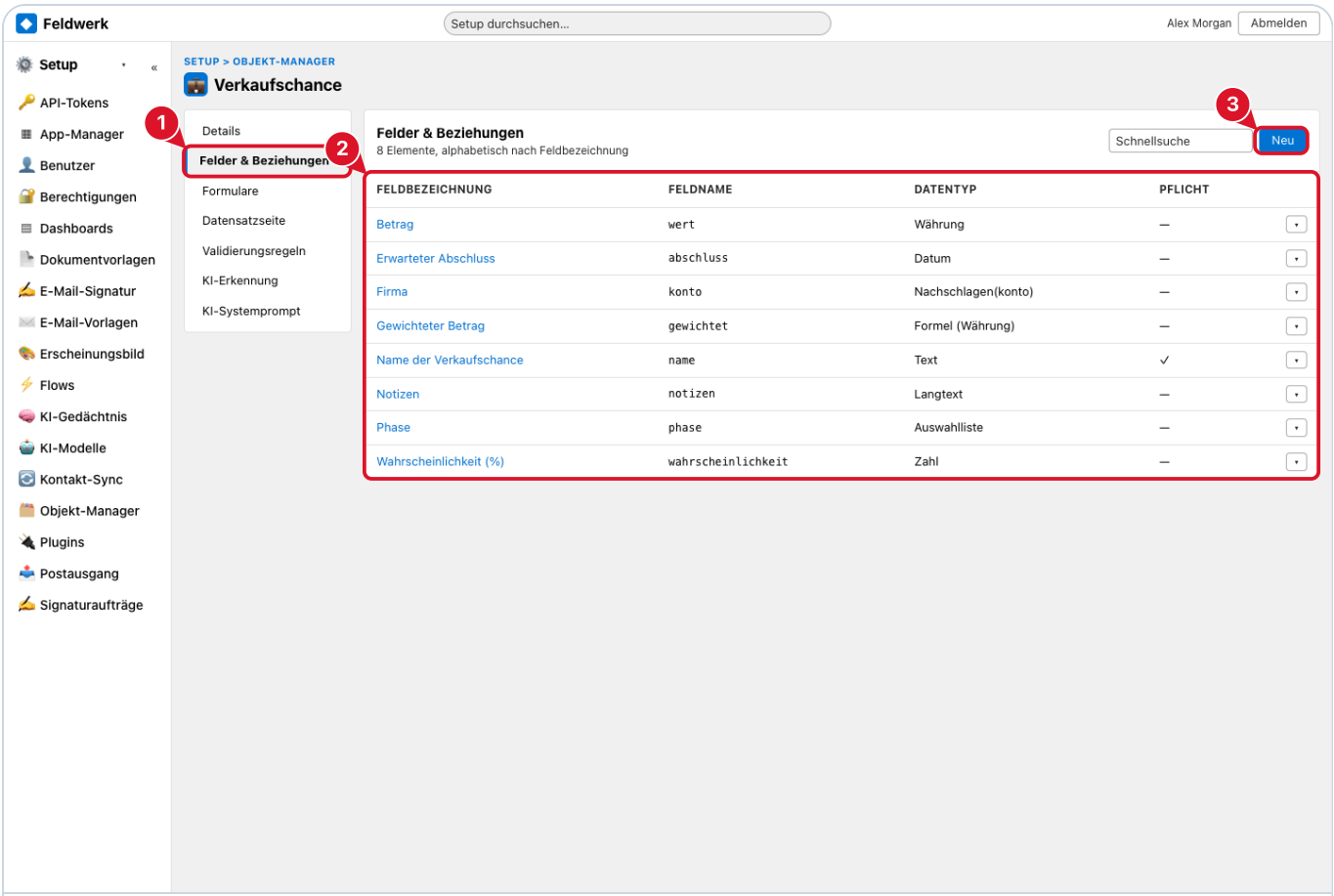
Ein Objekt ist eine Datensatzart, etwa Firma, Kontakt oder Rechnung. Ein Feld definiert einen Wert auf diesen Datensätzen. Entscheiden Sie vor dem Hinzufügen eines Objekts, wofür ein einzelner Datensatz steht und wie er mit bestehenden Objekten zusammenhängt. Erstellen Sie keine getrennten Objekte nur für verschiedene Zustände derselben Arbeitsart.

Administratoren verwalten diese Struktur unter **Setup → Objekt-Manager**. Benutzer arbeiten über Apps, Listen und Datensatzseiten mit den daraus entstehenden Datensätzen. Mit dem Anlegen eines Objekts ist die Benutzeroberfläche noch nicht vollständig eingerichtet: Prüfen Sie auch Formulare, Seite, App-Navigation und Berechtigungen.

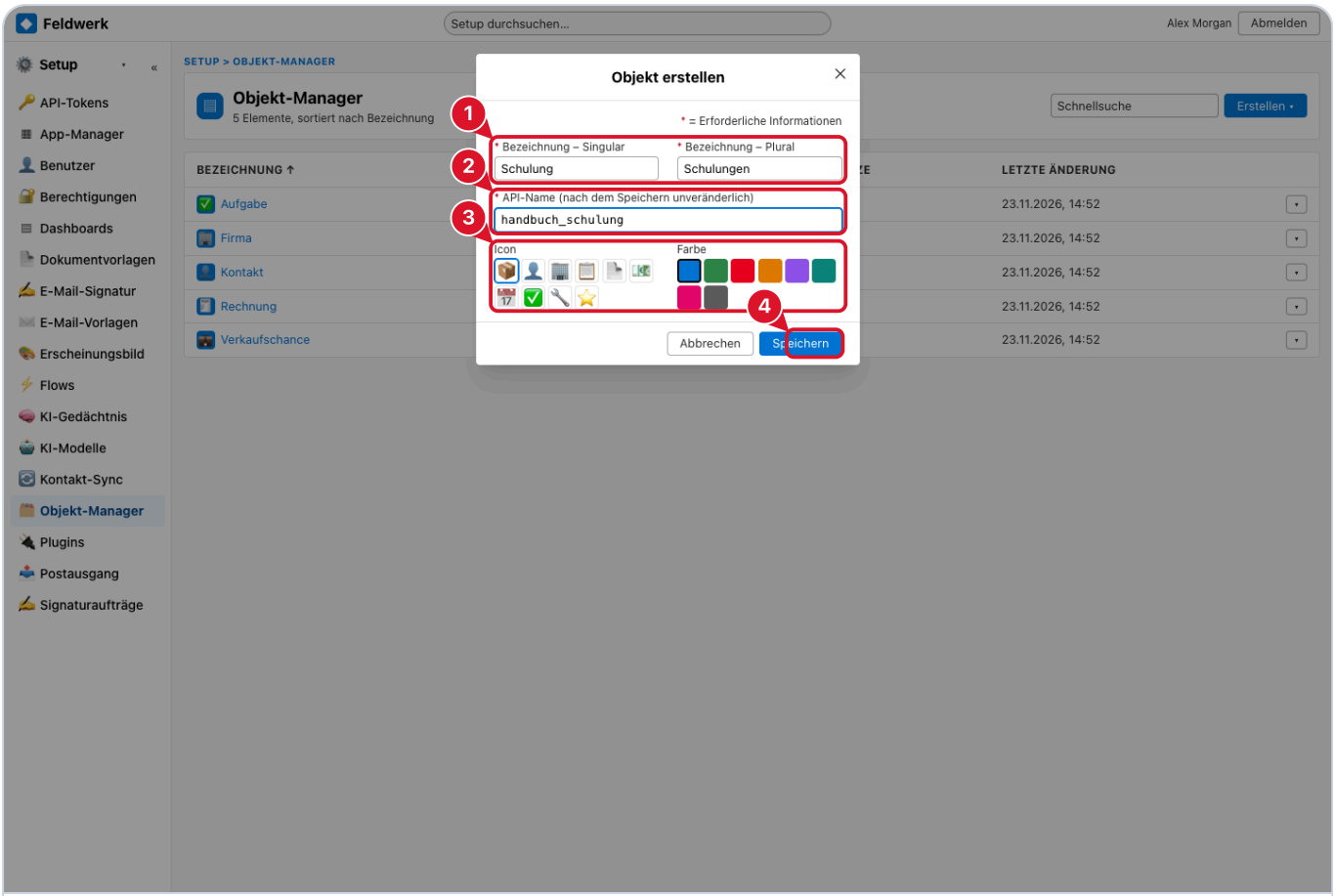
Ein Objekt anlegen

1. Öffnen Sie den Objekt-Manager und wählen Sie **Erstellen**.
2. Wählen Sie verständliche Beschriftungen für Einzahl und Mehrzahl.
3. Prüfen Sie vor dem Speichern den API-Namen. Er ist die stabile technische Kennung für Formeln und Integrationen.
4. Wählen Sie die verfügbaren Darstellungseinstellungen und speichern Sie.
5. Prüfen Sie Details sowie Felder und Beziehungen des neuen Objekts.

Die Beschriftung kann auf die Zielgruppe abgestimmt werden; der API-Name ist eine technische Vereinbarung. Im Handbuch gehört die Beschriftung **Verkaufschancen** zum API-Namen `chance`. Ersetzen Sie diesen Namen in einer Formel oder Integration nicht durch die Beschriftung.



1. Felder und Beziehungen des Objekts öffnen.
2. Beschriftungen, API-Namen und Feldtypen vergleichen.
3. Ein neues Feld hinzufügen.



1. Beschriftungen für Einzahl und Mehrzahl eingeben.
2. Den dauerhaften API-Namen prüfen.
3. Symbol und Farbe wählen.
4. Das neue Objekt speichern.

Die Benennung von Datensätzen festlegen

Wählen Sie in den **Details** des Objekts **Bearbeiten** und prüfen Sie den **Anzeigenamen**. Wählen Sie ein geeignetes Feld, etwa Schulungscode, damit Nachschlagefelder und Datensatzlinks einen erkennbaren Namen anzeigen. Die automatische Einstellung verwendet die ersten beiden Textfelder. Reicht ein Feld nicht aus, kann eine Textformel Werte kombinieren. Dieser Anzeigename ersetzt nicht die stabile öffentliche ID des Datensatzes.

Vorhandene Nachschlagewerte enthalten einen gespeicherten Anzeigenamen-Schnappschuss. Mit **Anzeigenamen prüfen** untersuchen Sie veraltete Verweise; mit **Anzeigenamen neu berechnen** aktualisieren Sie diese bei Bedarf. Das Ändern der Beschriftung oder des Anzeigenamens ändert nicht den Zieldatensatz der Beziehung.

Ein Feld anlegen

1. Wählen Sie das Objekt und öffnen Sie **Felder und Beziehungen**.
2. Wählen Sie **Neu** und den Feldtyp.
3. Geben Sie die Beschriftung ein und prüfen Sie den erzeugten API-Namen.
4. Legen Sie typspezifische Optionen, gegebenenfalls Pflichtfeld- oder Eindeutigkeitsregeln sowie einen hilfreichen Hilfetext fest.
5. Speichern Sie und prüfen Sie das Feld im Formular und auf der Datensatzseite des Objekts.
6. Erstellen Sie einen Testdatensatz, speichern Sie typische Werte und prüfen Sie die Darstellung.

Der API-Name lässt sich nach der Erstellung nicht mehr ändern. Auch der Feldtyp ist danach unveränderlich. Bei Nachschlagefeldern stehen Zielmodus und Menge der Zielobjekte nach der Erstellung fest. Planen Sie diese Entscheidungen, bevor Sie Geschäftsdaten speichern.

Den passenden Feldtyp wählen

Typ	Verwendung
Text / Langtext	Kurze Namen und Codes oder längere Notizen. Wählen Sie eine passende Länge.
Zahl / Währung / Prozent	Numerische Werte mit Genauigkeit und angezeigten Dezimalstellen, soweit unterstützt. Ein formatierter Betrag bleibt numerisch.
Datum / Datum-Uhrzeit	Ein Kalendertag oder ein Wert, der zusätzlich eine Uhrzeit enthält. Entscheidend ist die fachliche Bedeutung.
Checkbox	Ein Ja/Nein-Zustand.
Auswahlliste / Mehrfachauswahlliste	Vorgegebene Auswahlmöglichkeiten. Jede Option besitzt eine angezeigte Beschriftung und einen gespeicherten API-Wert.

Typ	Verwendung
E-Mail / Telefon / URL	Kontakt- und Adresswerte mit passender Eingabe und Darstellung.
Nachschlagebeziehung	Ein Link auf einen Datensatz. Der Zielmodus legt fest, welche Objekte ausgewählt werden können.
Objektreferenz	Ein Verweis auf eine Objektdefinition, nicht auf einen einzelnen Datensatz. Nützlich für Konfigurationen.
Formel	Ein berechneter schreibgeschützter Wert aus einem Ausdruck.
Aggregat	Ein berechneter Wert über verwandte untergeordnete Datensätze.
Autonummer	Ein automatisch vergebener formatierter Sequenzwert.
Unterschrift	Eine über die unterstützte Feldeingabe erfasste Unterschrift.
Datei/Bild	Ein zu einem Feld gehörender Dateiwert. Er unterscheidet sich vom allgemeinen Dateibereich des Datensatzes.

Pflichtwerte, Eindeutigkeit und Standardwerte

Ein Pflichtfeld muss beim Speichern eines Datensatzes ausgefüllt sein. Ein eindeutiges Feld verhindert, dass derselbe zulässige Wert in einem anderen Datensatz dieses Objekts erneut verwendet wird. Testen Sie beide Verhaltensweisen mit typischen Daten, bevor Sie sie in einem bestehenden Arbeitsablauf einsetzen.

Ein Standardwert ergänzt beim Anlegen einen fehlenden oder leeren Anfangswert; er ist kein fortlaufend neu berechnetes Feld. Beispielsweise erhält ein neuer Datensatz für Ort den Standardwert Online. Wenn Sie den Standardwert auf Workshopraum ändern, wird Online in einem bestehenden Datensatz bei einer späteren Bearbeitung nicht ersetzt. Eine in den Felddetails konfigurierte Standardwertformel kann Werte wie das heutige Datum liefern und hat Vorrang vor einem statischen Standardwert, sofern sie einen Wert ergibt. Verwenden Sie

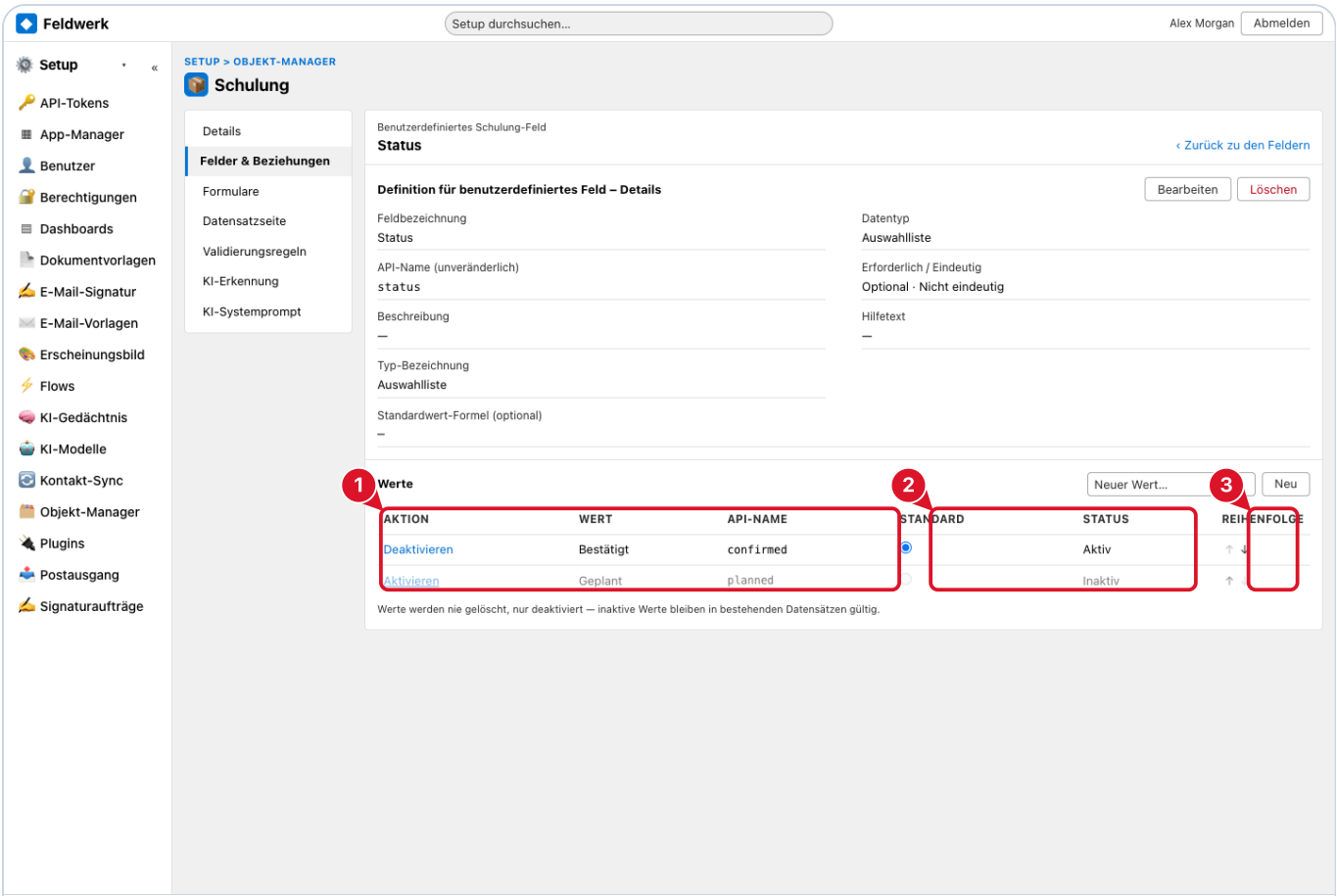
ein Formelfeld, wenn der Wert dauerhaft aus anderen Werten abgeleitet werden soll. Formel-, Aggregat- und Autonummernfelder lassen sich nicht als normale bearbeitbare Eingaben befüllen.

Auswahlmöglichkeiten pflegen

Halten Sie den gespeicherten Optionswert stabil, auch wenn sich die lesbare Beschriftung ändert. Formeln und Automatisierungen können diesen Wert verwenden. Prüfen Sie aktive Optionen, Standardoptionen und deren Reihenfolge. Vorhandene Auswahllistenoptionen können auch dann nicht gelöscht werden, wenn sie derzeit unbenutzt sind. Deaktivieren Sie veraltete Optionen stattdessen.

1. Öffnen Sie die Detailseite des Auswahllistenfelds und suchen Sie die Tabelle **Werte**.
2. Wählen Sie mit dem Standard-Auswahlknopf eine aktive Standardoption.
3. Ändern Sie die Optionsreihenfolge mit den Auf-/Ab-Schaltflächen.
4. Wählen Sie **Deaktivieren**, um eine Option stillzulegen, oder **Aktivieren**, um sie wieder verfügbar zu machen. Beim Deaktivieren der Standardoption wird auch ihre Standardkennzeichnung entfernt.
5. Geben Sie zum Hinzufügen einer Option ihre Beschriftung in die Eingabe für einen neuen Wert ein und wählen Sie **Neu**. Prüfen Sie den erzeugten API-Wert.

Diese Aktionen speichern sofort. Bestehende Datensatzwerte werden nicht allein deshalb auf eine andere Option umgeschrieben, weil eine Option deaktiviert wurde. Halten Sie bei der Beschriftungspflege über die API den gespeicherten API-Wert stabil.



1. Optionen unter Beibehaltung ihrer API-Werte aktivieren oder deaktivieren.
2. Einen aktiven Standardwert wählen.
3. Die Anzeigereihenfolge ändern.

Strukturen vorsichtig ändern

Eine neue Feldbeschriftung benennt die API-Kennung nicht um. Eine Formularänderung ändert nur das Eingabelayout, nicht den Feldtyp. Prüfen Sie Verweise in Formeln, Berichten, Flows, Dokumenten und Integrationen, bevor Sie ein Feld oder Objekt entfernen.

Das Löschen von Strukturen kann gespeicherte Informationen entfernen. Arbeiten Sie zunächst in einer Testinstanz, prüfen Sie Abhängigkeiten und sichern Sie benötigte Geschäftsdaten vor destruktiven Änderungen. Ein in der Navigation verborgenes Objekt oder Feld kann weiterhin existieren und an anderer Stelle verwendet werden.

Ein Feld löschen

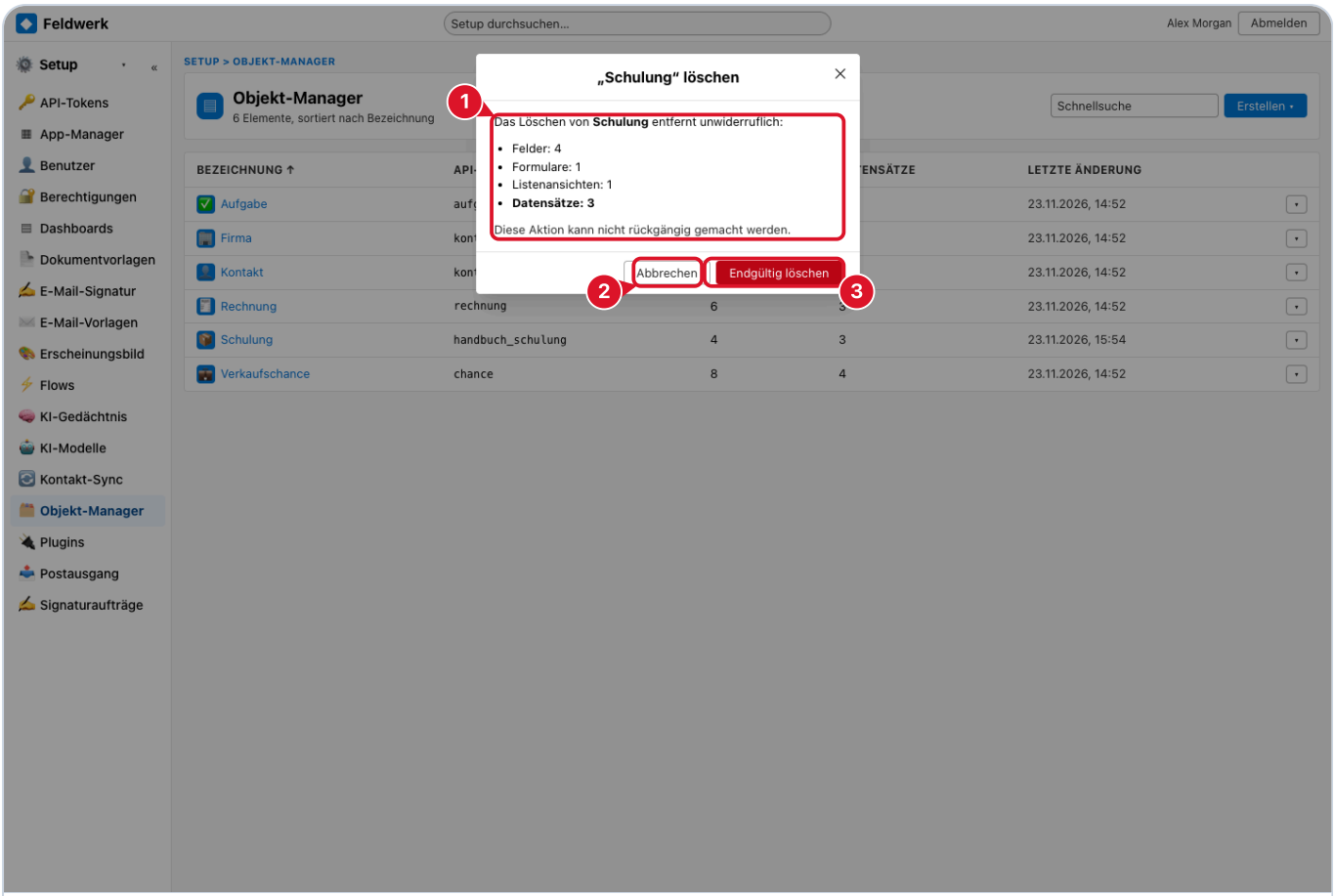
Ein Pflichtfeld ist vor Löschung geschützt. Machen Sie es nur dann zuerst optional, wenn diese Änderung beabsichtigt ist. Auch Verweise aus Flows, Berichten, Formeln und Aggregaten können das Löschen verhindern. Lösen Sie die gemeldeten Abhängigkeiten vor

einem erneuten Versuch. Eine erfolgreiche Feldlöschung entfernt dessen gespeicherte Werte aus den Datensätzen des Objekts und seine Verweise aus Formularen und Listenlayouts. Das ist nicht dasselbe wie das Ausblenden eines Felds im Formular.

Ein Objekt löschen

1. Öffnen Sie im Objekt-Manager das Aktionsmenü des Objekts und wählen Sie **Löschen**.
2. Prüfen Sie im Bestätigungsdialog die Anzahl der Felder, Formulare, Listenansichten und Datensätze.
3. Wählen Sie **Abbrechen**, falls Sie etwas behalten oder exportieren müssen. Wählen Sie **Endgültig löschen** nur, wenn die Entfernung beabsichtigt ist.

Ein eingehendes Nachschlagefeld aus einem anderen Objekt kann die Löschung verhindern, selbst wenn noch kein Datensatz es verwendet. Interne Objekte, die einem Plugin gehören, lassen sich so nicht löschen. Eine erfolgreiche Löschung entfernt die Datensätze und Struktur des Objekts einschließlich Formularen, Seiten, Listen, Berichten, Flows, Validierungsregeln, Anhängen und App-Registern. Offene polymorphe Beziehungen an anderer Stelle können nach dem Entfernen des Zielobjekts einen veralteten Anzeigenamen-Schnappschuss behalten; er verweist dann nicht mehr auf einen vorhandenen Datensatz. Für die Objektlöschung gibt es kein normales Rückgängigmachen.



1. Anzahl und dauerhafte Wirkung prüfen.
2. Mit Abbrechen das Objekt behalten.
3. Nur bestätigen, wenn die aufgeführten Daten und Strukturen entfernt werden sollen.

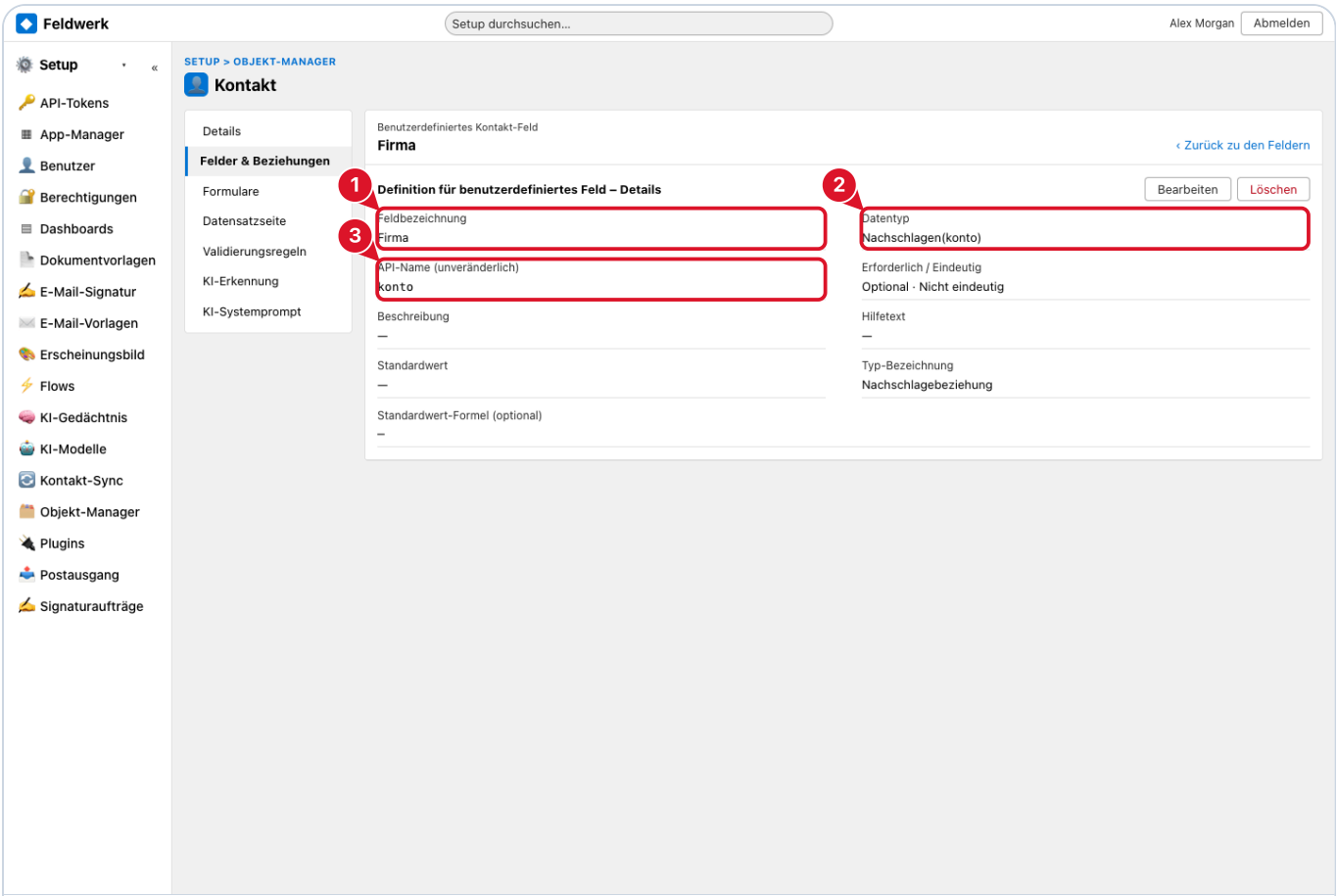
Nachschlagefelder und verwandte Listen

Datensätze mit Nachschlagefeldern verbinden und Beziehungen in Seiten, Formeln und Berichten nutzen.

Was ein Nachschlagefeld speichert

Ein Nachschlagefeld verweist auf einen anderen Datensatz. Beispielsweise hat ein Kontakt ein Nachschlagefeld Firma. Viele Kontakte können auf dieselbe Firma verweisen; deren verwandte Liste kann diese Kontakte anzeigen. Die Beziehung gehört zum Nachschlagefeld auf dem untergeordneten Objekt.

Verwenden Sie ein Nachschlagefeld, wenn das Ziel ein Datensatz mit eigener Identität und eigenen Daten ist. Ein Firmenname in einem einfachen Textfeld erstellt nicht dieselbe Beziehung. Auch eine Objektreferenz ist etwas anderes: Sie identifiziert eine Objektdefinition, keinen einzelnen Datensatz.



1. Die Feldbeschriftung wird Benutzern angezeigt.
2. Das Nachschlageziel gehört zur Felddefinition.
3. In Konfigurationen den stabilen API-Namen verwenden.

Das Ziel konfigurieren

1. Wählen Sie im Objekt-Manager das untergeordnete Objekt und öffnen Sie Felder und Beziehungen.
2. Erstellen Sie ein Feld vom Typ **Nachschlagebeziehung**.
3. Wählen Sie ein **Zielobjekt**. Der aktuelle Feldassistent erstellt eine Beziehung zu genau einem festen Objekt.
4. Legen Sie Beschriftung, API-Namen und weitere unterstützte Eigenschaften fest und speichern Sie.
5. Platzieren Sie das Feld im passenden Formular und prüfen Sie die zugehörige Konfiguration der verwandten Liste auf der übergeordneten Seite.

Die Plattform unterstützt auch eine feste Menge von Zielobjekten oder beliebige erlaubte Objekte. Diese Beziehungen zu mehreren Objekten werden über die reguläre API konfiguriert, nicht über den aktuellen Feldassistenten. Um Firmen und Kontakte zuzulassen, verwenden Sie deren tatsächliche API-Namen in der Felddefinition:

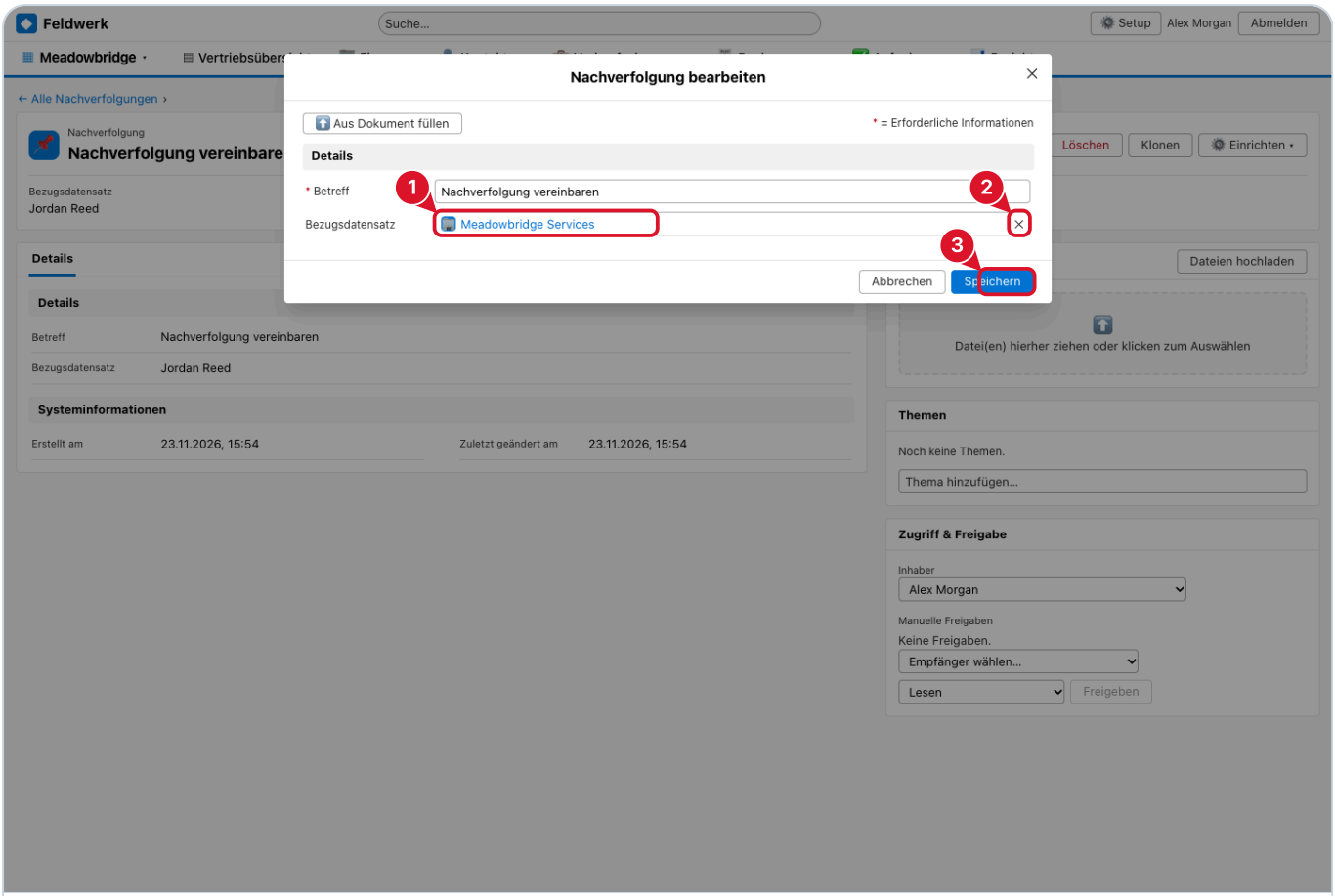
```
"optionen": {
  "zielObjekte": ["konto", "kontakt"]
}
```

Verwenden Sie genau einen Zielmodus: `zielObjekt` für ein Objekt, `zielObjekte` für eine feste Menge oder `zielBeliebig: true` für eine offene Beziehung. Bei mehreren möglichen Zielen müssen Benutzer auch das Zielobjekt wählen. Zielmodus und Zielmenge sind nach der Erstellung unveränderlich. Erstellen Sie eine neue Beziehung und planen Sie eine Migration, wenn das Modell geändert werden muss.

Einen verknüpften Datensatz wählen

1. Bearbeiten Sie den Quelldatensatz.
2. Wählen Sie bei einem Nachschlagefeld für mehrere Objekte zuerst das gewünschte Objekt.
3. Geben Sie einen charakteristischen Teil des Datensatznamens ein und wählen Sie den richtigen Vorschlag.
4. Speichern Sie und prüfen Sie den verknüpften Wert auf der Datensatzseite.

Nutzen Sie weitere Erkennungsmerkmale, wenn mehrere Vorschläge denselben Namen haben. Leeren Sie die Auswahl, um eine optionale Beziehung zu entfernen, oder wählen Sie einen anderen Datensatz als Ersatz. Ein Pflicht-Nachschlagefeld darf auch beim Bearbeiten eines bestehenden Datensatzes nicht leer bleiben. Das Speichern wird abgelehnt, bis Sie ein zulässiges Ziel gewählt haben. Auch eine offene Beziehung listet nur zugängliche Objekte und umgeht keine Objekt- oder Datensatzberechtigungen.



1. Gewählten Datensatz und Objektsymbol prüfen. Dieses Feld erlaubt Firmen oder Kontakte.
2. Auswahl leeren, um die Beziehung zu entfernen oder zu ersetzen.
3. Änderung speichern; im Hintergrund steht noch der zuvor gespeicherte Kontakt.

Der Beziehung folgen

Wählen Sie den Nachschlagelink, um den verknüpften Datensatz zu öffnen. Eine konfigurierte verwandte Liste am übergeordneten Datensatz zeigt die darauf verweisenden untergeordneten Datensätze. Ein Beziehungsfilter ändert den Umfang einer Liste; er erstellt keinen neuen Datenbestand.

Im Handbuch verweist Serviceplanung auf Meadowbridge Services. Die verwandte Aufgabenliste verwendet das Nachschlagefeld Verkaufschance an Aufgaben. Das Feld Rechnungssumme der Firma folgt dem Nachschlagefeld Firma an Rechnungen, um sein Aggregat zu berechnen.

Nachschlagepfade verwenden

Formeln können unterstützten Nachschlagepfaden über Feld-API-Namen folgen, etwa `konto.name` für den Namen der verknüpften Firma. Die Feldauswahl im Formeleditor hilft bei der Wahl eines gültigen Pfads. Fehlende Beziehungen und nicht zugängliche Daten erfordern eine bewusste Leerwertbehandlung. Testen Sie diese Fälle ebenso wie vollständig ausgefüllte Datensätze.

In Flow-Bedingungen verwenden Datensatzfelder die Flow-Kontextschreibweise, etwa `$Datensatz.konto`. Vermischen Sie diese Schreibweise nicht mit einer einfachen Feldformel, ohne den Ausdruckskontext zu prüfen.

Zugriff und Lebenszyklus prüfen

Eine Beziehung erteilt keine Berechtigung. Benutzer benötigen weiterhin den passenden Zugriff, um das Ziel auszuwählen oder zu öffnen. Ist eine Beziehung nicht auswählbar, prüfen Sie Zielkonfiguration, Berechtigungen und die Existenz des Ziels.

Prüfen Sie vor dem Löschen eines referenzierten Datensatzes die eingehenden Beziehungen und die Auswirkungen auf verwandte Listen, Formeln und Automatisierungen. Gehen Sie nicht davon aus, dass alle untergeordneten Datensätze mit dem übergeordneten gelöscht werden. Prüfen Sie nach einer beabsichtigten Löschung die verbleibenden Datensätze und passen Sie Geschäftsabläufe an, die von der Beziehung abhingen.

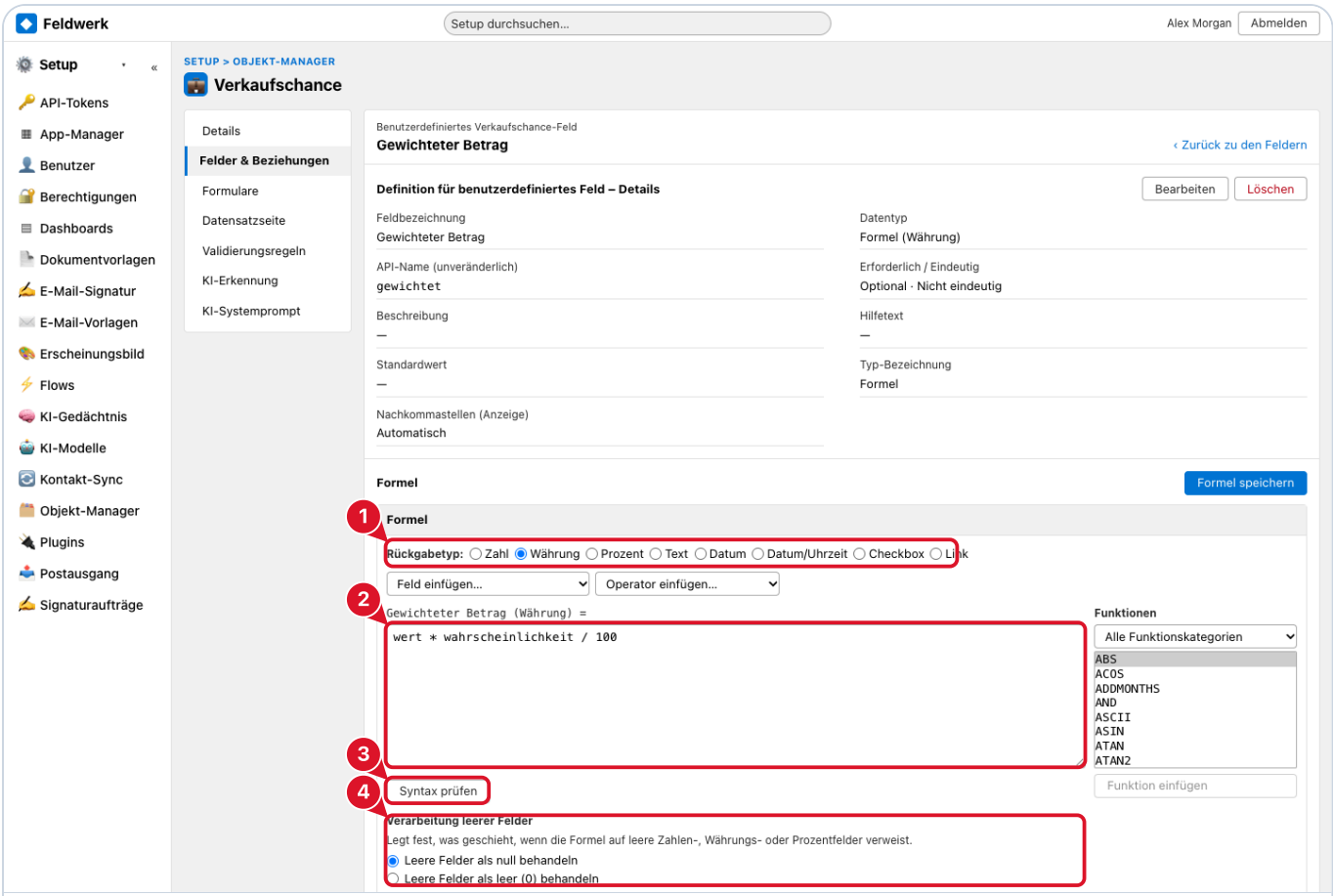
Formelfelder

Werte aus Datensatzfeldern berechnen: mit typisierten Ausdrücken, bewusster Leerwertbehandlung und einer geprüften Funktionsreferenz.

Eine Formel statt eines kopierten Werts wählen

Ein Formelfeld leitet seinen Wert beim Lesen des Datensatzes ab. In normalen Formularen ist es schreibgeschützt. Verwenden Sie es, wenn das Ergebnis seinen Eingangswerten folgen soll. Soll ein Wert dagegen zu einem bestimmten Ereignis zugewiesen oder gespeichert werden, verwenden Sie einen Standardwert oder eine Automatisierung.

Das Handbuchfeld Gewichteter Betrag verwendet `wert * wahrscheinlichkeit / 100`. Dabei steht `wert` für Betrag und `wahrscheinlichkeit` für das Zahlenfeld Wahrscheinlichkeit (%). Mit 18000 und 60 ergibt sich 10800. Wenn Sie die Wahrscheinlichkeit auf 50 ändern, ändert sich das berechnete Ergebnis auf 9000.



1. Ergebnistyp wählen.
2. Stabile Feld-API-Namen im Ausdruck verwenden.
3. Vor dem Speichern die Syntax prüfen.
4. Verhalten leerer numerischer Eingaben festlegen.

Eine Formel anlegen und prüfen

1. Öffnen Sie Felder und Beziehungen des Objekts und erstellen Sie ein Formelfeld.
2. Wählen Sie den Rückgabetyp: Zahl, Währung, Prozent, Text, Datum, Datum/Uhrzeit, Checkbox oder Link.
3. Geben Sie den Ausdruck ein. Nutzen Sie bei Bedarf Feld einfügen, Operator einfügen und die Funktionsliste.
4. Legen Sie fest, wie leere numerische Felder behandelt werden.
5. Wählen Sie **Syntax prüfen**, korrigieren Sie Fehler und wählen Sie im Assistenten für ein neues Feld **Speichern**. Bei einem vorhandenen Feld wählen Sie **Formel speichern**.
6. Prüfen Sie Datensätze mit normalen Werten, Leerwerten und Grenzwerten.

Eine erfolgreiche Syntaxprüfung beweist nicht, dass die fachliche Berechnung korrekt ist. Prüfen Sie bekannte Eingaben und berechnen Sie das erwartete Ergebnis unabhängig. Kontrollieren Sie den deklarierten Rückgabetyp und die angezeigten Dezimalstellen. Die Syntaxprüfung meldet die aus dem Ausdruck abgeleitete Ergebnisfamilie: Die Addition zweier Währungseingaben kann Zahl melden, obwohl der deklarierte Rückgabetyp des Felds Währung bleibt. Der deklarierte Typ bestimmt die Feldausgabe.

Grundlagen der Ausdrücke

Verwenden Sie Feld-API-Namen statt Beschriftungen. Unterstützte Rechenoperatoren sind `+`, `–`, `*`, `/` und `^`; Klammern steuern die Gruppierung. Mit `&` verketten Sie Text. Vergleiche umfassen =, <>, <, >, <= und >=. Textliterale stehen in Anführungszeichen; Funktionsargumente werden durch Kommas getrennt.

Verknüpfte Werte erreichen Sie über unterstützte Nachschlagepfade wie `konto.name`. Formelfeldausdrücke verwenden diesen Feldkontext. Flow-Ausdrücke haben ihren eigenen Kontext, unter anderem `$Datensatz`. Prüfen Sie den Kontext, bevor Sie Ausdrücke kopieren.

Leerwerte und ungültige Berechnungen

Die Leerwertbehandlung legt fest, ob leere Eingaben vom Typ Zahl, Währung und Prozent leer bleiben oder als null behandelt werden. Verwenden Sie `ISBLANK`, `BLANKVALUE` oder einen bedingten Ausdruck, wenn die fachliche Bedeutung einen ausdrücklichen Ersatzwert verlangt.

Eine Division durch null liefert ein leeres Ergebnis. Auch ein fehlender Nachschlagewert kann einen abgeleiteten Wert leer lassen. Verbergen Sie diese Fälle nicht hinter einer beliebigen Null, sofern Null nicht fachlich korrekt ist. Verweise auf andere Berechnungen dürfen keine Zyklen erzeugen.

Für einen konkreten Leerwerttest erstellen Sie die Währungseingaben `betrag` und `zuschlag` sowie die Währungsformel `betrag + zuschlag`. Bei Betrag 100 und leerem Zuschlag ergibt **Leere Felder als Nullen behandeln** den Wert 100; **Leere Felder als leer behandeln** ergibt ein leeres Ergebnis. Mit Zuschlag 25 entsteht 125. Der Versuch, das Formelergebnis direkt über die API zuzuweisen, wird abgelehnt.

Feld einfügen setzt den API-Namen eines Felds an die Cursorposition. Operator einfügen setzt das gewählte Symbol ein. Wählen Sie eine Funktionskategorie und eine Funktion und klicken Sie auf **Funktion einfügen**, um ihren Namen mit leeren runden Klammern

einzusetzen. Ergänzen Sie die Argumente vor der Syntaxprüfung. Nachschlagepfade können Sie manuell eingeben. Auch nach erfolgreicher lokaler Syntaxprüfung sind beim Speichern die abschließende Pfad- und Zyklusprüfung des Servers erforderlich.

Ausgeführte Beispiele

Diese Beispiele wurden mit derselben Formel-Engine ausgeführt, die Feldwerk verwendet.

Ausdruck	Ergebnis
`ROUND(18.456, 2)`	18.46
`IF(2 > 1, "Ready", "Waiting")`	Ready
`UPPER("Meadowbridge")`	MEADOWBRIDGE
`FIXED(2400, 2)`	2400.00
`FORMATNUMBERDE(2400, 2)`	2.400,00
`DATE(2026, 12, 15)`	2026-12-15
`YEAR(DATE(2026, 12, 15))`	2026

Funktionen nach Zweck wählen

- **Logik:** IF und CASE wählen Werte; AND, OR und NOT verknüpfen Bedingungen. ISPICKVAL vergleicht den gespeicherten Auswahllistenwert.
- **Zahlen:** ROUND und TRUNC steuern Dezimalstellen; MIN und MAX wählen Extremwerte. CEILING/FLOOR und MCEILING/MFLOOR unterscheiden sich bei negativen Werten: CEILING(-3.2) ist -4, MCEILING(-3.2) dagegen -3.
- **Text:** LEFT, RIGHT, MID und FIND arbeiten mit Teilzeichenfolgen und Positionen. FIND unterscheidet Groß- und Kleinschreibung und liefert eine bei eins beginnende Position oder null, wenn nichts gefunden wird. REGEXEXTRACT liefert den ersten passenden Text oder die erste Gruppe; ein ungültiges Muster oder fehlender Treffer ergibt leeren Text. REGEXTEST liefert bei einem ungültigen Muster false.
- **Formatierung:** FIXED erzeugt feste Dezimalstellen mit Punkt und ohne Tausendertrennzeichen. FORMATNUMBERDE erzeugt deutsche Zahlenformatierung. Diese Funktionsnamen sind stabile sprachunabhängige Kennungen.

- **Datum:** TODAY und NOW verwenden die aktuelle Zeit der Engine. YEAR, MONTH und DAY entnehmen Datumsbestandteile; ADDMONTHS verschiebt um Kalendermonate und begrenzt einen überlaufenden Tag auf den letzten Tag des Zielmonats. DATETIMEVALUE interpretiert Text ohne Zeitzone als UTC. LOCALTEXT formatiert Datum/Uhrzeit in Europe/Berlin.
- **Links und Kodierung:** HYPERLINK liefert einen Linkwert. RECORDURL erzeugt aus einem Nachschlagewert eine Datensatz-URL. URLENCODE und XMLENCODE kodieren Werte für den jeweiligen Ausgabekontext.
- **Anhänge:** ATTACHMENTCOUNT liefert die Anhanganzahl des aktuellen Datensatzes; HASATTACHMENTS zeigt an, ob Anhänge vorhanden sind.

Kompatibilitätsgrenzen

IMAGE liefert seinen Alternativtext, statt ein Bild einzubetten. CASESAFEID gibt den Eingabetext zurück, ohne eine ID umzuwandeln. HYPERLINK akzeptiert ein optionales Zielargument, ignoriert es jedoch. TIMENOW liefert derzeit einen Datum/Uhrzeit-Wert. FORMATDURATION formatiert eine absolute Dauer in ganzen Sekunden als Stunden:Minuten:Sekunden; sein optionales zweites Argument wird nicht verwendet.

CURRENCYRATE, DISTANCE, GEOLOCATION, GETSESSIONID und PICKLISTCOUNT sind verfügbar. Wechselkurse benötigen einen aktuellen Anbieter-Schnappschuss; die Sitzungskennung benötigt eine angemeldete Browsersitzung. Beispiele und Grenzen finden Sie auf den einzelnen Funktionsseiten.

Verfügbare Funktionen: Syntax und Beispiele

Wählen Sie eine Funktion, um die Bedeutung jedes Arguments, optionale Argumente und das Ergebnis des Beispiels nachzulesen. Neue Formeln verwenden immer englische Funktionsnamen, auch in der deutschen Oberfläche. Vorhandene deutsche Namen bleiben als Kompatibilitätsaliase gültig. Feld-API-Namen und Textwerte bleiben unverändert. Die vollständige Referenz ist auch in der PDF enthalten.

Zahlen und Mathematik

Funktion und Syntax	Zweck
`ABS(number)`	Liefert den Betrag einer Zahl ohne Vorzeichen.

Funktion und Syntax	Zweck
`ACOS(number)`	Liefert den Arkuskosinus als Winkel im Bogenmaß.
`ASIN(number)`	Liefert den Arkussinus als Winkel im Bogenmaß.
`ATAN(number)`	Liefert den Arkustangens als Winkel im Bogenmaß.
`ATAN2(y, x)`	Liefert den Winkel eines zweidimensionalen Vektors im Bogenmaß.
`CEILING(number)`	Rundet eine Zahl von null weg auf eine ganze Zahl.
`COS(angle)`	Liefert den Kosinus eines Winkels.
`DISTANCE(location1, location2, unit)`	Berechnet die Großkreisentfernung zwischen zwei Koordinatenpaaren.
`EXP(number)`	Potenziert die mathematische Konstante e mit dem angegebenen Exponenten.
`FLOOR(number)`	Rundet eine Zahl in Richtung null auf eine ganze Zahl.
`GEOLOCATION(latitude, longitude)`	Erzeugt ein Koordinatenpaar für eine DISTANCE-Berechnung.
`LN(number)`	Liefert den natürlichen Logarithmus (Basis e).
`LOG(number)`	Liefert den Logarithmus zur Basis 10.
`MAX(number1, number2, ...)`	Liefert den größten angegebenen numerischen Wert.
`MCEILING(number)`	Rundet in Richtung positiv unendlich auf.
`MFLOOR(number)`	Rundet in Richtung negativ unendlich ab.

Funktion und Syntax	Zweck
`MIN(number1, number2, ...)`	Liefert den kleinsten angegebenen numerischen Wert.
`MOD(dividend, divisor)`	Liefert den Rest einer Division.
`PI()`	Liefert die mathematische Konstante Pi.
`PICKLISTCOUNT(selection)`	Zählt die gewählten Werte einer Mehrfachauswahlliste.
`ROUND(number, decimal_places)`	Rundet eine Zahl auf die angegebene Anzahl von Dezimalstellen.
`SIN(angle)`	Liefert den Sinus eines Winkels.
`SQRT(number)`	Liefert die nicht negative Quadratwurzel.
`TAN(angle)`	Liefert den Tangens eines Winkels.
`TRUNC(number [, decimal_places])`	Schneidet Dezimalstellen ohne Rundung ab.

Logik und Bedingungen

Funktion und Syntax	Zweck
`AND(condition1, condition2, ...)`	Liefert nur dann true, wenn jede Bedingung wahr ist.
`BLANKVALUE(value, fallback)`	Ersetzt einen fehlenden Wert oder leeren Text durch einen Ersatzwert.
`CASE(value, match1, result1, match2, result2, ..., fallback)`	Vergleicht einen Wert mit Alternativen und liefert das erste passende Ergebnis.
`IF(condition, value_if_true, value_if_false)`	Wählt anhand einer Bedingung zwischen zwei Werten.
`ISBLANK(value)`	Prüft, ob ein Wert fehlt oder eine leere Zeichenfolge ist.

Funktion und Syntax	Zweck
`ISNULL(value)`	Prüft, ob ein Wert fehlt oder eine leere Zeichenfolge ist.
`ISNUMBER(value)`	Prüft, ob ein Wert numerisch ist oder als Zahl gelesen werden kann.
`ISPICKVAL(picklist, stored_value)`	Vergleicht eine einfache Auswahlliste mit einem gespeicherten API-Wert.
`NOT(condition)`	Kehrt eine Wahrheitswertbedingung um.
`NULLVALUE(value, fallback)`	Ersetzt einen fehlenden Wert oder leeren Text durch einen Ersatzwert.
`OR(condition1, condition2, ...)`	Liefert true, wenn mindestens eine Bedingung wahr ist.

Text und Links

Funktion und Syntax	Zweck
`ASCII(text)`	Liefert die UTF-16-Codeeinheit des ersten Zeichens.
`BEGINS(text, prefix)`	Prüft, ob ein Text mit einer bestimmten Zeichenfolge beginnt.
`BR()`	Fügt einen Zeilenumbruch in Text ein.
`CHR(code)`	Erzeugt ein UTF-16-Zeichen aus einem numerischen Code.
`CONTAINS(text, search_text)`	Prüft, ob ein Text eine Teilzeichenfolge enthält.
`FIND(search_text, text [, start])`	Findet die erste Position einer Teilzeichenfolge.

Funktion und Syntax	Zweck
`FIXED(number [, decimal_places])`	Formatiert eine Zahl als Text mit festen Dezimalstellen, Dezimalpunkt und ohne Gruppierung.
`FORMATNUMBERDE(number [, decimal_places])`	Formatiert eine Zahl als deutschen Zahlentext mit Tausendergruppierung und Dezimalkomma.
`HYPERLINK(url, label [, target])`	Erzeugt einen Linkwert mit Zieladresse und Anzeigetext.
`IMAGE(url, alternative_text [, height, width])`	Liefert den Alternativtext eines Bilds; diese Implementierung stellt kein Bild dar.
`INCLUDES(choices, stored_value)`	Prüft auf einen exakten Eintrag in einem Mehrfachauswahlwert.
`INITCAP(text)`	Wandelt Text in Kleinbuchstaben und Wortanfänge in Großbuchstaben um.
`LEFT(text, count)`	Entnimmt Zeichen vom Textanfang.
`LEN(text)`	Zählt die Zeichen eines Texts.
`LOWER(text)`	Wandelt Text in Kleinbuchstaben um.
`LPAD(text, length [, padding])`	Füllt Text links bis zur gewünschten Länge auf.
`MID(text, start, count)`	Entnimmt Zeichen ab einer Position innerhalb eines Texts.
`RECORDURL(lookup)`	Liefert die relative Feldwerk-URL eines referenzierten Datensatzes.
`REGEXEXTRACT(text, pattern [, group])`	Extrahiert den ersten Treffer eines regulären Ausdrucks oder eine erfasste Gruppe.
`REGEXTEST(text, pattern)`	Prüft, ob ein regulärer Ausdruck auf einen Teil des Texts passt.

Funktion und Syntax	Zweck
`REVERSE(text)`	Kehrt die Reihenfolge der Textzeichen um.
`RIGHT(text, count)`	Entnimmt Zeichen vom Textende.
`RPAD(text, length [, padding])`	Füllt Text rechts bis zur gewünschten Länge auf.
`SUBSTITUTE(text, old_text, new_text)`	Ersetzt jedes Vorkommen einer Teilzeichenfolge.
`TEXT(value)`	Wandelt einen Wert in Text um.
`TRIM(text)`	Entfernt Leerraum an beiden Textenden.
`UPPER(text)`	Wandelt Text in Großbuchstaben um.
`URLENCODE(text)`	Kodiert einen einzelnen URL-Bestandteil.
`VALUE(text)`	Liest numerischen Text als Zahl ein.
`XMLENCODE(text)`	Maskiert XML-relevante Zeichen in einem Textwert.

Datum und Uhrzeit

Funktion und Syntax	Zweck
`ADDMONTHS(date_or_datetime, months)`	Verschiebt ein Datum um eine Anzahl von Kalendermonaten.
`DATE(year, month, day)`	Erzeugt ein Datum aus drei Zahlen.
`DATETIMEVALUE(value)`	Liest einen Zeitstempel ein oder wandelt ein typisiertes Datum in Datum/Uhrzeit um.
`DATEVALUE(value)`	Wandelt ISO-Datumstext oder typisiertes Datum/Uhrzeit in ein Datum um.
`DAY(date_or_datetime)`	Liest den Tag des Monats aus einem Datum.

Funktion und Syntax	Zweck
`DAYOFYEAR(date_or_datetime)`	Liest die Tagesnummer innerhalb des Jahres (1–366) aus einem Datum.
`FORMATDURATION(seconds [, compatibility_option])`	Formatiert eine verstrichene Dauer als Stunden:Minuten:Sekunden.
`FROMUNIXTIME(seconds)`	Wandelt Unix-Epochensekunden in Datum/Uhrzeit um.
`HOUR(date_or_datetime)`	Liest die Stunde (0–23) aus einem Datum.
`ISOWEEK(date_or_datetime)`	Liest die ISO-Kalenderwochennummer aus einem Datum.
`ISOYEAR(date_or_datetime)`	Liest das ISO-Kalenderwochenjahr aus einem Datum.
`LOCALTEXT(date_or_datetime)`	Formatiert Datum/Uhrzeit als Text in der Ortszeit Europe/Berlin.
`MILLISECOND(date_or_datetime)`	Liest die Millisekunde (0–999) aus einem Datum.
`MINUTE(date_or_datetime)`	Liest die Minute (0–59) aus einem Datum.
`MONTH(date_or_datetime)`	Liest den Monat (1–12) aus einem Datum.
`NOW()`	Liefert das aktuelle Datum mit Uhrzeit.
`SECOND(date_or_datetime)`	Liest die Sekunde (0–59) aus einem Datum.
`TIMENOW()`	Liefert das aktuelle Datum mit Uhrzeit, in dieser Implementierung wie NOW.
`TIMEVALUE(date_or_datetime)`	Liefert aus einem vorhandenen typisierten Datum oder Datum/Uhrzeit einen typisierten Datum/Uhrzeit-Wert.
`TODAY()`	Liefert das heutige Datum in UTC.
`UNIXTIMESTAMP(date_or_datetime)`	Wandelt Datum/Uhrzeit in ganze Sekunden seit der Unix-Epoche um.

Funktion und Syntax	Zweck
`WEEKDAY(date_or_datetime)`	Liest den Wochentag (1 = Sonntag, 7 = Samstag) aus einem Datum.
`YEAR(date_or_datetime)`	Liest das Jahr aus einem Datum.

Datensatzkontext und Kompatibilität

Funktion und Syntax	Zweck
`ATTACHMENTCOUNT()`	Liefert die Anzahl der Anhänge des aktuellen Datensatzes.
`CASESAFEID(value)`	Liefert die Eingabe als Text, ohne die Kennung umzuwandeln.
`CURRENCYRATE(currency_code)`	Liefert Einheiten der Unternehmenswährung je Einheit der angefragten Währung.
`GETSESSIONID()`	Liefert die öffentliche Kennung der aktuellen angemeldeten Browsersitzung.
`HASATTACHMENTS()`	Prüft, ob der aktuelle Datensatz mindestens einen Anhang hat.

Kontextabhängige Funktionen

Alle Funktionen dieses Katalogs sind implementiert. GETSESSIONID benötigt eine angemeldete Browsersitzung. CURRENCYRATE benötigt einen verfügbaren aktuellen Kurs des konfigurierten Anbieters. Die Detailseiten erklären leere Ergebnisse beziehungsweise null, wenn der Kontext fehlt.

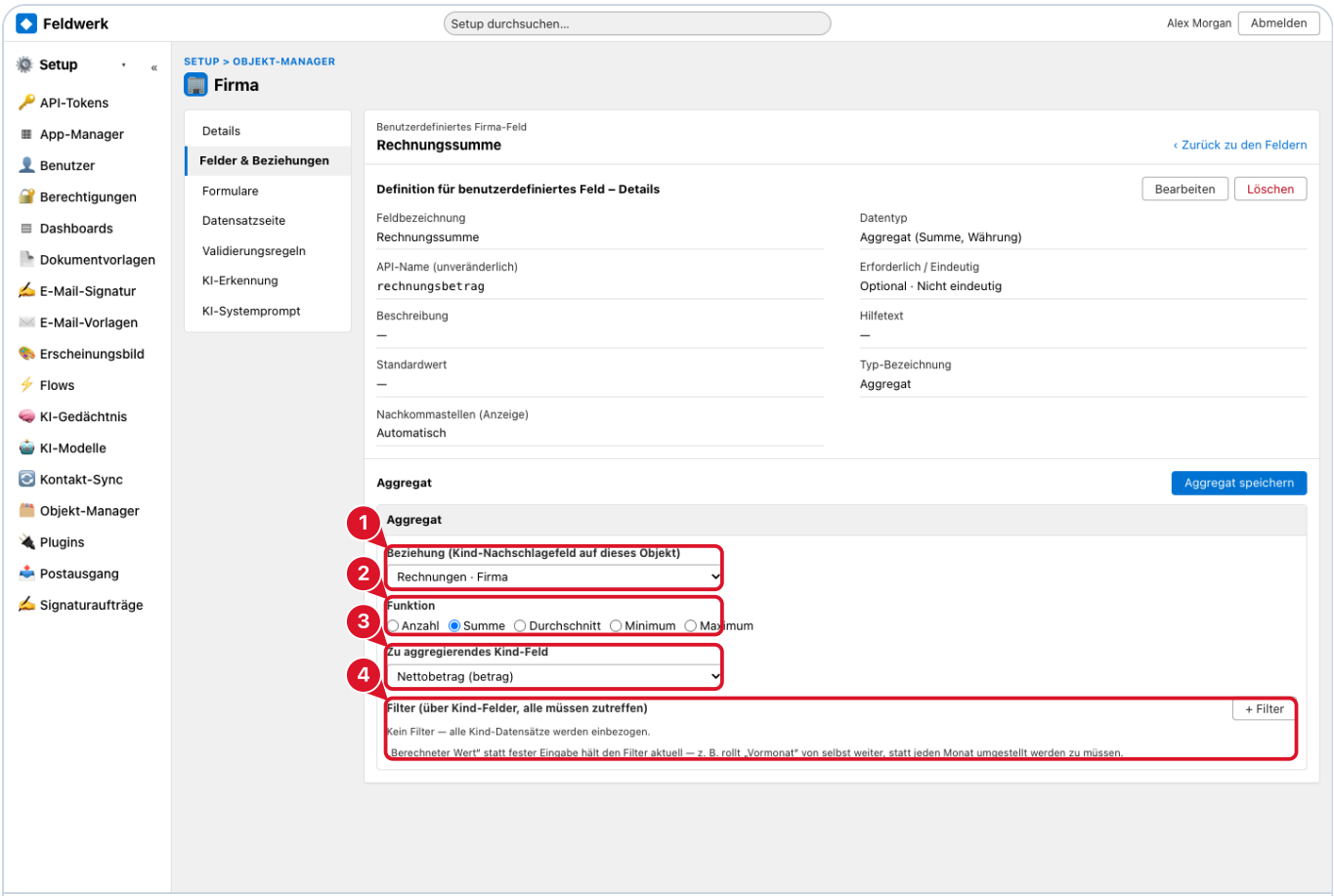
Aggregat- und Autonummernfelder

Verwandte Datensätze zusammenfassen und Kennungen ohne manuelles Rechnen oder Eingeben vergeben.

Untergeordnete Datensätze zusammenfassen

Ein Aggregatfeld gehört zum übergeordneten Objekt und berechnet Werte über untergeordnete Datensätze, die durch ein Nachschlagefeld darauf verweisen. Wählen Sie die Beziehung sorgfältig: Das Feld am untergeordneten Objekt bestimmt, welche Datensätze einfließen.

Unser Firmenobjekt enthält die **Rechnungssumme**. Das Feld folgt **Rechnungen · Firma**, summiert das untergeordnete Feld **Nettobetrag** und zeigt ein Währungsergebnis. Die beiden Meadowbridge-Rechnungen liefern 2400 und 6000, insgesamt 8400.



1. Untergeordnete Beziehung wählen.
2. Aggregatfunktion wählen.
3. Wertfeld des untergeordneten Objekts wählen.
4. Beteiligte Datensätze mit optionalen Filtern begrenzen.

Ein Aggregat konfigurieren

1. Öffnen Sie Felder und Beziehungen des übergeordneten Objekts und erstellen Sie ein Aggregatfeld.
2. Wählen Sie die auf dieses Objekt verweisende untergeordnete Nachschlagebeziehung.
3. Wählen Sie Anzahl, Summe, Durchschnitt, Minimum oder Maximum.
4. Wählen Sie für Funktionen, die einen Wert benötigen, ein kompatibles Feld des untergeordneten Objekts.
5. Ergänzen Sie Filter auf untergeordnete Felder, wenn nur bestimmte verwandte Datensätze einfließen sollen.
6. Wählen Sie **Aggregat speichern** und prüfen Sie das Ergebnis an einem übergeordneten

Datensatz.

Anzahl zählt passende Datensätze. Summe addiert die gewählten Werte. Durchschnitt, Minimum und Maximum fassen das gewählte kompatible Feld zusammen. Ihr Ergebnistyp hängt von Eingabe und Funktion ab. Aggregatwerte werden beim Lesen von Datensätzen berechnet und sind schreibgeschützt; ändern Sie die beitragenden Datensätze, um das Ergebnis zu ändern. Anzahl berücksichtigt passende Datensätze auch dann, wenn das ausgewählte Wertfeld leer wäre. Die übrigen Funktionen ignorieren Leerwerte: Die Beträge 100, 200 und leer ergeben Summe 300 und Durchschnitt 150. Ohne verwendbare passende Werte bleibt das Ergebnis leer statt null; Anzahl liefert für eine leere Menge null. Minimum und Maximum unterstützen auch kompatible Datumsfelder.

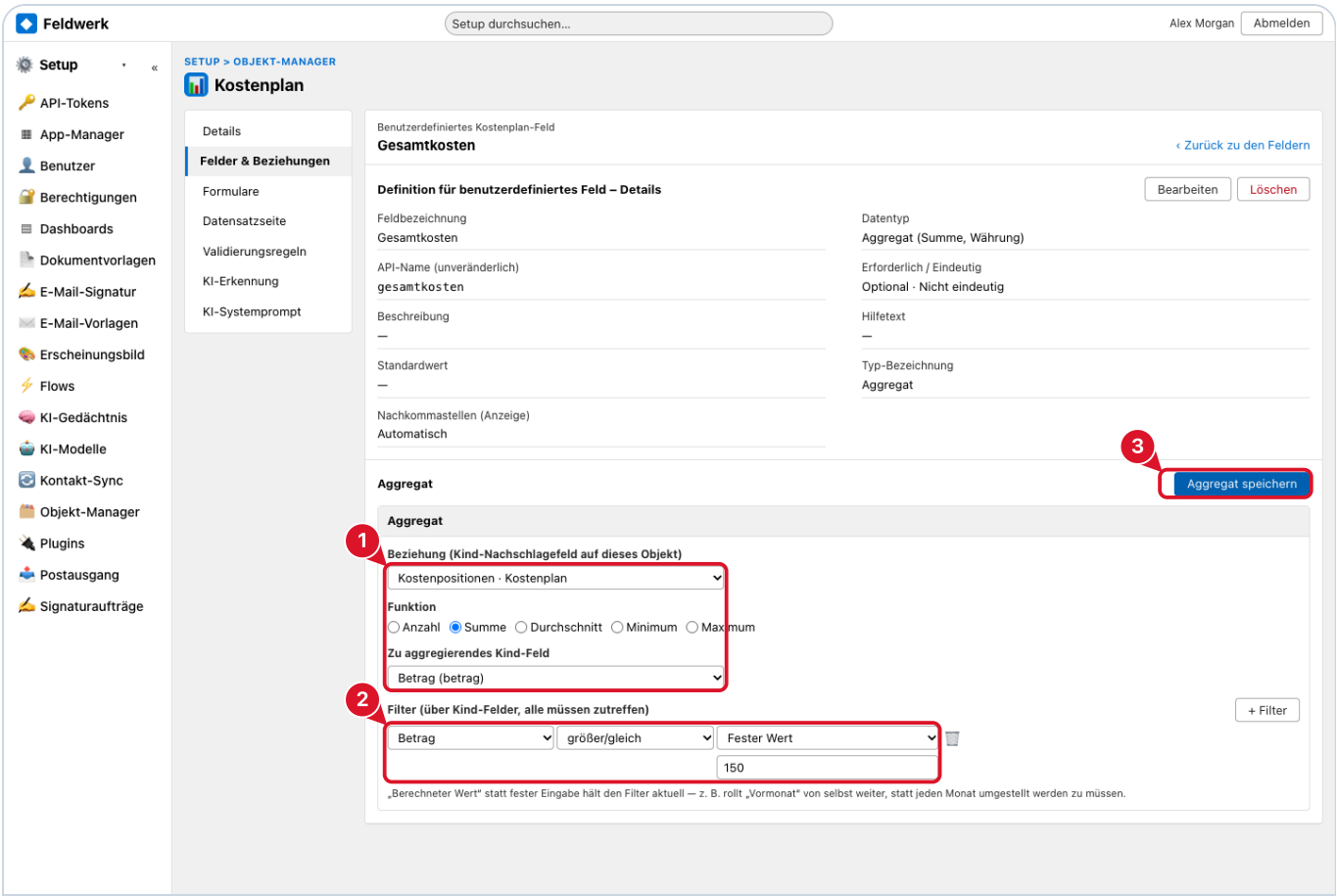
Beitragende Datensätze filtern

Filter gelten für untergeordnete Felder; alle konfigurierten Bedingungen müssen zutreffen. Ohne Filter fließen alle untergeordneten Datensätze der Beziehung ein. Nutzen Sie für einen mitwandernden Datumsbereich einen unterstützten berechneten oder relativen Wert, statt jeden Monat ein festes Datum manuell zu ändern.

Testen Sie einen passenden und einen nicht passenden untergeordneten Datensatz, einen geänderten Wert und eine entfernte Beziehung. Im Handbuchtest erhöhen zwei neue verknüpfte Rechnungen über jeweils 100 die Rechnungssumme von 8400 auf 8600; nach dem Löschen dieser Testrechnungen beträgt sie wieder 8400.

Ein Beispiel mit gefilterten Kosten

Das isolierte Beispiel **Kostenpläne** enthält drei verwandte Kostenpositionen mit den Beträgen 100, 200 und leer. Wählen Sie **Summe**, **Betrag** und den Filter **Betrag ist größer oder gleich 150**. Die Summe beträgt dann 200. Ändern Sie die beitragende Position auf 250, wird die Summe ebenfalls 250. Wenn Sie ihr Nachschlagefeld Kostenplan leeren oder die Position löschen, bleibt die Summe leer, da keine Position mehr passt.



1. Untergeordnete Beziehung, Summe und Betrag wählen.
2. Nur Beträge ab 150 einbeziehen.
3. Konfiguration speichern und den übergeordneten Datensatz erneut lesen.

Wählen Sie nach Änderungen an Funktion oder Filtern **Aggregat speichern** und aktualisieren Sie anschließend den übergeordneten Datensatz. Ein Wechsel der ausgewählten Beziehung leert das Wertfeld und die Filter des Entwurfs; wählen Sie diese für die neue Beziehung erneut.

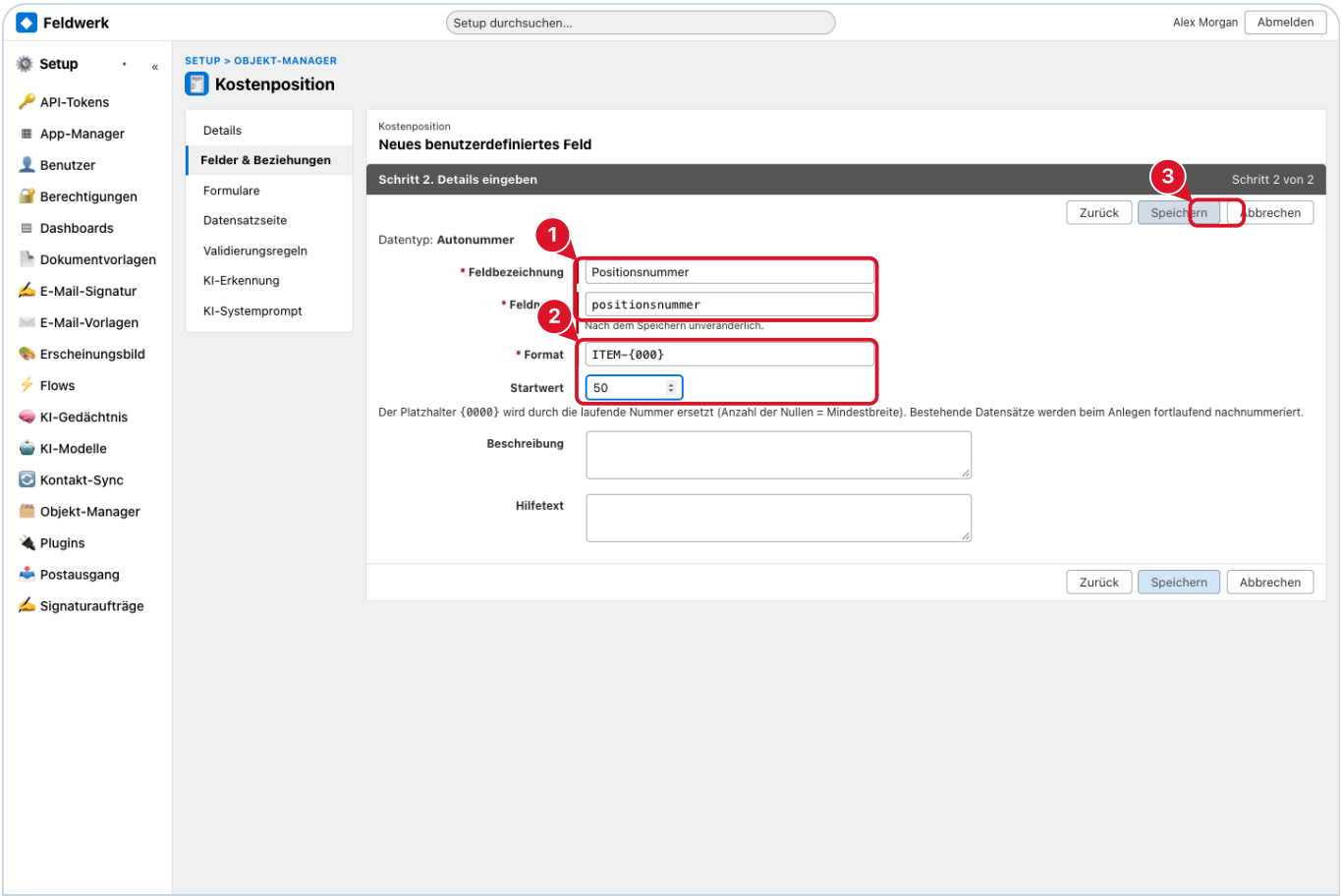
Automatische Nummern vergeben

Ein Autonummernfeld vergibt einen formatierten Sequenzwert. Es ist keine bearbeitbare Datensatzeingabe. Die Beispiel-Rechnungsnummer verwendet `INV-{0000}` mit dem Startwert `1001` und erzeugt Werte wie `INV-1001`.

1. Erstellen Sie am Objekt ein Autonummernfeld.
2. Wählen Sie ein Format mit genau einem Nullplatzhalter, etwa `{0000}`.
3. Legen Sie den ersten Sequenzwert fest und speichern Sie das Feld.

4. Erstellen Sie zwei Testdatensätze und prüfen Sie, dass sie unterschiedliche aufeinanderfolgende Nummern erhalten.

Der Nullplatzhalter legt die Mindestbreite mit führenden Nullen fest. Feldwerk vergibt die Sequenz; Integrationen und Benutzer dürfen keinen eigenen Ersatzwert liefern. Wenn Sie das Feld zu einem Objekt mit bestehenden Datensätzen hinzufügen, erhalten auch diese Nummern. Prüfen Sie deshalb den vorgesehenen Startwert, bevor Sie dies auf Geschäftsdaten anwenden.



1. Feld benennen; sein API-Name bleibt nach der Erstellung unveränderlich.
2. Einen Nullplatzhalter verwenden und die erste Nummer wählen.
3. Speichern; auch bestehende Datensätze erhalten Nummern.

Im gezeigten Kostenpositionen-Test erhalten die beiden bestehenden Datensätze mit `ITEM-{000}` und Startnummer 50 die Werte `ITEM-050` und `ITEM-051`. Ein neuer Datensatz erhält `ITEM-052`. Nach dessen Löschung bekommt der nächste neue Datensatz `ITEM-053`. Eine über die API mitgelieferte Ersatznummer überschreibt weder den erzeugten noch den bereits vergebenen Wert.

Das Sequenzverhalten verstehen

Automatische Nummern sind Kennungen und keine Anzahl der aktuellen Datensätze. Eine Datensatzlöschung gibt dessen Nummer nicht zur normalen Wiederverwendung frei. Auch Testdatensätze verbrauchen Nummern. Eine Lücke beweist daher keinen Datenverlust; die höchste Nummer sagt nicht, wie viele Datensätze noch vorhanden sind.

Eine formatierte Geschäftsnummer unterscheidet sich von der internen beziehungsweise öffentlichen Identität des Datensatzes. Integrationen sollten zum Adressieren eines bestimmten Datensatzes die unterstützte Datensatz-ID verwenden. Eine neue Anzeigebeschriftung ändert diese Identität nicht.

Unerwartete Ergebnisse prüfen

- **Falsche Aggregatsumme:** Prüfen Sie Nachschlagefeld, gewähltes Wertfeld und Filter der untergeordneten Datensätze und aktualisieren Sie den übergeordneten Datensatz.
- **Keine Beziehung angeboten:** Erstellen oder prüfen Sie das untergeordnete Nachschlagefeld, das auf dieses übergeordnete Objekt verweist.
- **Schreibgeschützter Wert:** Bearbeiten Sie Quellwerte oder Konfiguration, nicht das berechnete Feld.
- **Nummer weicht vom Beispiel ab:** Lokaler Sequenzstand und Startwert bestimmen sie; die Handbuchnummern dienen der Veranschaulichung.

Validierungsregeln

Ungültige Änderungen verhindern und hilfreiche Korrekturhinweise anzeigen.

Was eine Validierungsregel bewirkt

Eine Validierungsregel prüft einen Datensatz beim Speichern. Ihre Formel beschreibt den ungültigen Fall: Ist das Ergebnis wahr, lehnt Feldwerk den Speichervorgang ab und zeigt die konfigurierte Meldung. Das unterscheidet sich von einem Pflichtfeld, das lediglich einen Wert verlangt, und von einer Automatisierung, die einen Wert verändert.

Administratoren konfigurieren Regeln am Objekt. Personen, die Datensätze bearbeiten, sehen die daraus entstehenden Meldungen, ohne Zugriff auf das Setup zu benötigen. Regeln gelten auch für Änderungen über die reguläre API oder über Automatisierungen, die den Datensatzdienst verwenden.

Beispiel: einen negativen Betrag ablehnen

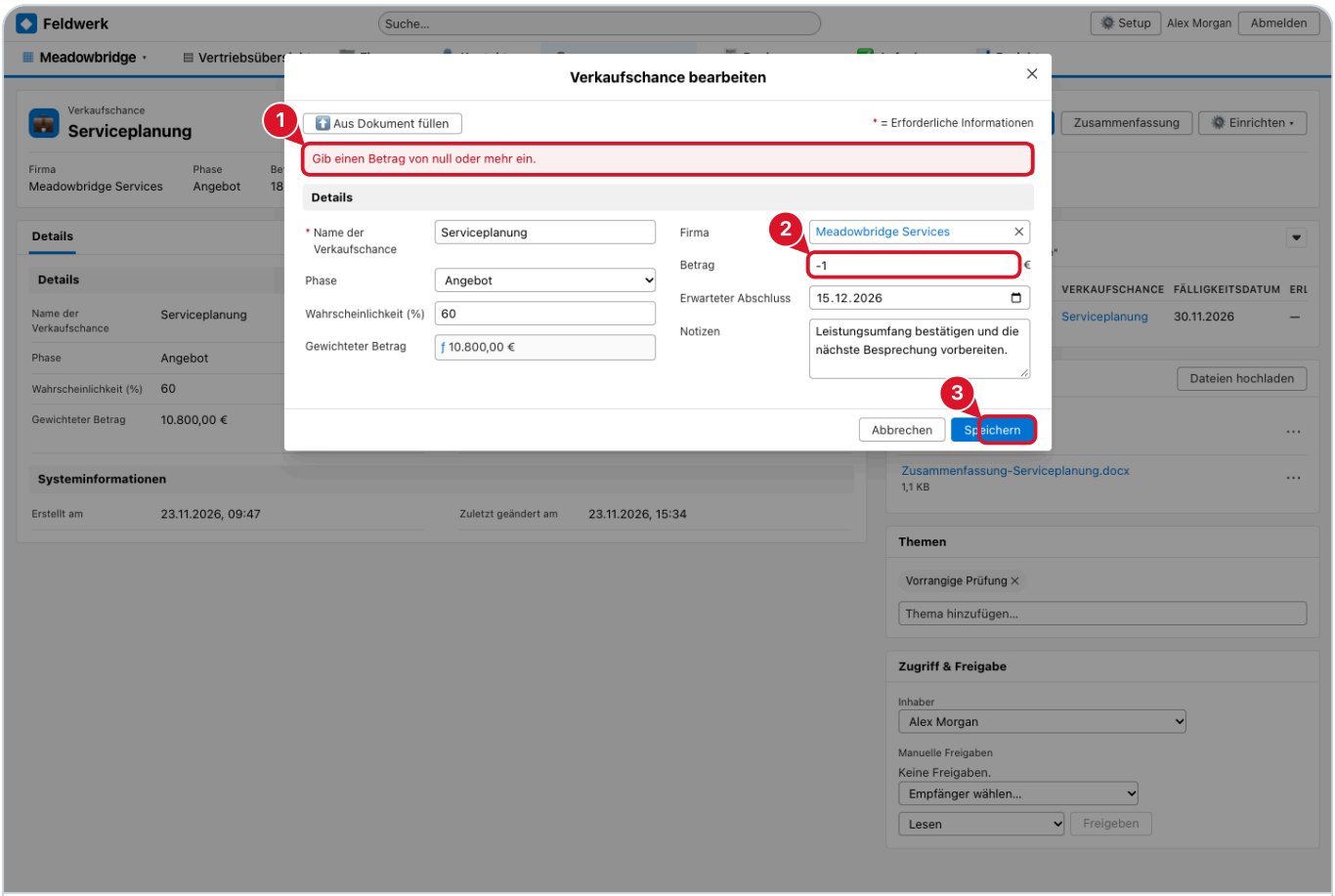
Unser Objekt Verkaufschancen hat ein Betragsfeld mit dem API-Namen `wert`. Ein negativer Betrag ist in diesem Beispiel ungültig. Die Regelformel lautet `wert < 0`; die Meldung lautet **Geben Sie einen Betrag von null oder mehr ein.** Sie erklärt, was als Nächstes zu tun ist.

1. Öffnen Sie **Setup → Objekt-Manager** und wählen Sie das Objekt.
2. Öffnen Sie **Validierungsregeln** und wählen Sie **Neu**.
3. Benennen Sie die Regel so, dass andere Administratoren ihren Zweck erkennen.
4. Geben Sie die Formel für den ungültigen Fall mit den tatsächlichen Feld-API-Namen ein.
5. Geben Sie eine konkrete Fehlermeldung ein und speichern Sie die Regel.
6. Prüfen Sie, dass die Regel aktiv ist. Eine inaktive Regel schützt spätere Speichervorgänge nicht.

Verwenden Sie normale Feld-API-Namen in Validierungsformeln. Die Schreibweise `$Datensatz.field` aus Flow-Bedingungen gehört zum Flow-Kontext; übernehmen Sie sie nicht ungeprüft in diese Formel.

Beide Ergebnisse testen

1. Öffnen Sie eine Test-Verkaufschance und wählen Sie **Bearbeiten**.
2. Setzen Sie den **Betrag** auf `-1` und wählen Sie **Speichern**.
3. Prüfen Sie, dass der Dialog mit der konfigurierten Korrekturmeldung offen bleibt. Der ungültige Wert wurde nicht gespeichert.
4. Setzen Sie Betrag auf `18000`, speichern Sie erneut und prüfen Sie den Wert auf der Datensatzseite.

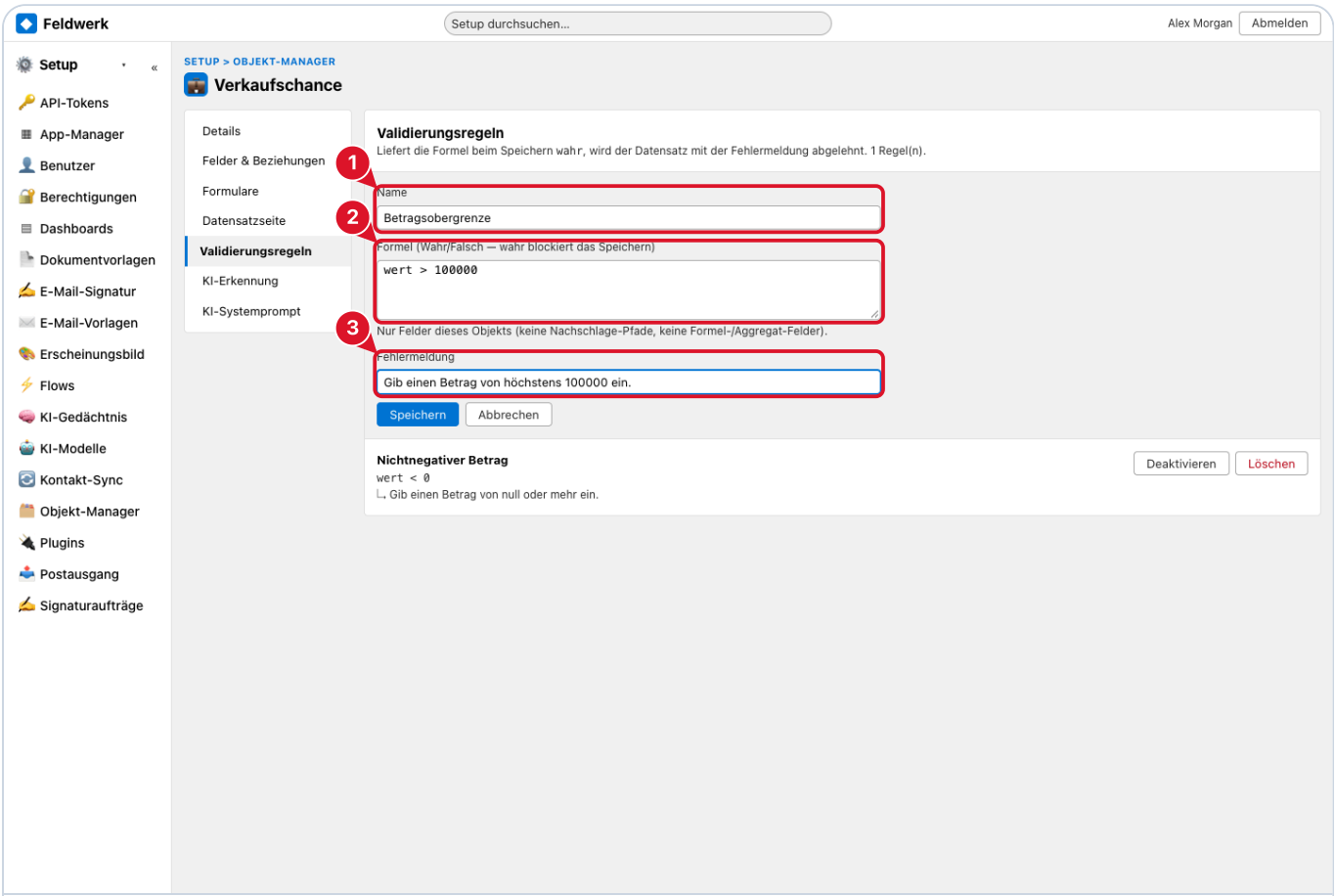


1. Konfigurierten Korrekturhinweis lesen.
2. Ungültigen Betrag ersetzen.
3. Korrigierten Datensatz speichern.

Im Beispiel liefert der negative Speicherversuch einen Validierungsfehler; der korrigierte Versuch ist erfolgreich. Testen Sie außerdem null, einen Leerwert, sofern das Feld ihn erlaubt, und weitere fachlich relevante Grenzwerte. Eine Formel, die jeden Datensatz ablehnt, kann die tägliche Arbeit unterbrechen.

Bei Bedarf eine Obergrenze hinzufügen

Eine separate Regel kann Beträge oberhalb einer fachlichen Grenze ablehnen. Die folgende Konfiguration verwendet `wert > 100000` mit der Meldung **Geben Sie einen Betrag von höchstens 100000 ein.** Genau `100000` ist erlaubt, `100001` wird abgelehnt. In unserem Beispiel bestehen auch null und ein leerer Betrag diese Obergrenzenregel. Sollen leere Beträge abgelehnt werden, verwenden Sie die Pflichtfeldeinstellung oder eine separate Regel.



1. Einen aussagekräftigen Namen vergeben.
2. Wahr bedeutet, dass das Speichern abgelehnt wird.
3. Erklären, wie der Wert zu korrigieren ist.

Handlungsorientierte Meldungen schreiben

Beschreiben Sie die Korrektur, nicht bloß die fehlgeschlagene Validierung. „Wählen Sie ein Enddatum am oder nach dem Startdatum“ hilft mehr als „Ungültiger Datensatz“. Konfigurierte Meldungen sind Inhalte Ihrer Organisation und bleiben in der Sprache, in der Sie sie geschrieben haben. Ein Wechsel der Oberflächensprache übersetzt sie nicht.

Eine Regel ändern oder deaktivieren

Verwenden Sie die Bedienelemente der Regel unter Validierungsregeln. Eine Deaktivierung beendet die künftige Durchsetzung, repariert aber keine bestehenden Daten. Dokumentieren Sie vor dem Löschen, weshalb die Regel existierte, und testen Sie den Ersatz. Blockiert eine unerwartete Regel Ihre Arbeit, nennen Sie der Administration Objekt, versuchte Änderung und genaue Meldung. Geben Sie niemals Zugangsdaten weiter.

Wenn die Meldung überrascht

Prüfen Sie zuerst die Richtung der Formel: wahr bedeutet ablehnen. Prüfen Sie anschließend Feld-API-Namen, Leerwerte und andere aktive Regeln desselben Objekts. Das Speichern kann außerdem konfigurierte Automatisierungen auslösen. Korrigieren Sie die Regel oder die zugrunde liegenden Daten, statt mit einem anderen Eingabebildschirm die Validierung umgehen zu wollen.

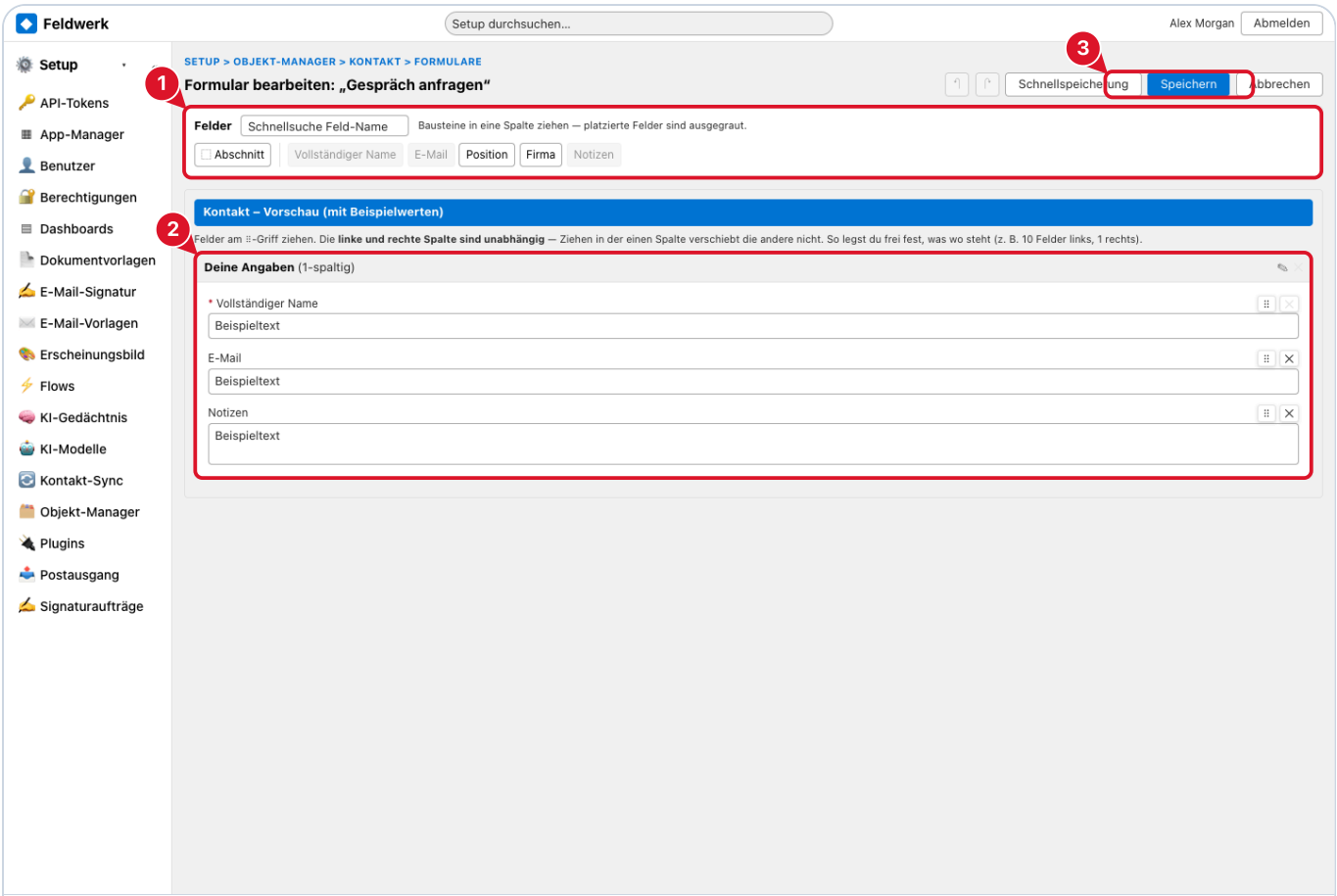
Formular-Builder

Datensatzeingaben in Abschnitten und Spalten anordnen und das gespeicherte Formular testen.

Formulare und Datensatzseiten haben unterschiedliche Aufgaben

Ein Formular ordnet die Felder für die Eingabe oder Bearbeitung von Werten an. Eine Datensatzseite ordnet Komponenten wie Details, verwandte Listen und Dateien an. Eine Formularänderung erstellt keine neuen Felder und ändert keine Objektberechtigungen.

Administratoren öffnen **Setup → Objekt-Manager → das Objekt → Formulare** und wählen beim Formular **Builder**. Die Kopfzeile des Builders zeigt, welches Layout bearbeitet wird. Prüfen Sie diesen Namen vor dem Speichern, wenn das Objekt mehrere Formulare besitzt.



1. Felder und Abschnittsbausteine wählen.
2. Felder in unabhängigen Spalten anordnen.
3. Formular speichern und mit echten Datensätzen prüfen.

Felder anordnen

1. Suchen Sie ein Feld in der Feldpalette, bei Bedarf über die Schnellsuche nach Feldnamen.
2. Ziehen Sie es in eine Abschnittsspalte. Bereits platzierte Felder sind in der Palette ausgegraut.
3. Verschieben Sie das Feld an seinem Ziehgriff an die gewünschte Position.
4. Prüfen Sie die Beispielvorschau und ordnen Sie die übrigen Felder an.

Linke und rechte Spalte sind unabhängig. Ein verschobenes Feld in einer Spalte verschiebt nicht das entsprechende Feld in der anderen. Ein langes Notizfeld kann deshalb neben mehreren kurzen Feldern stehen, ohne starre zeilenweise Paarung.

Abschnitte konfigurieren

Ziehen Sie einen Abschnittsbaustein in das Layout und öffnen Sie seine Eigenschaften. Wählen Sie einen Abschnittsnamen, die Sichtbarkeit seiner Überschrift sowie eine oder zwei Spalten. Legen Sie die Tab-Reihenfolge zeilenweise oder spaltenweise fest und bestätigen Sie die Eigenschaften.

Eine Änderung der Spaltenanzahl verteilt vorhandene Abschnittsinhalte neu. Prüfen Sie anschließend Reihenfolge und Tastaturnavigation. Gruppieren Sie Felder nach Benutzeraufgabe, etwa Kontaktdaten, statt nach technischen Feldtypen.

Pflichteingaben erreichbar halten

Pflichtfelder lassen sich nicht einfach aus dem Layout entfernen. Auch ein Abschnitt mit Pflichtfeldern ist gegen Entfernen geschützt. Soll ein Feld nicht länger Pflicht sein, prüfen Sie diese fachliche Regel in der Felddefinition, statt die Eingabe zu verstecken.

Das Entfernen eines optionalen Felds aus einem Layout löscht weder seine Definition noch bestehende Werte. Es schränkt auch den Feldzugriff nicht ein. Verwenden Sie zur Zugriffskontrolle Feldberechtigungen.

Speichern und prüfen

1. Prüfen Sie letzte Layoutänderungen mit Rückgängig und Wiederholen.
2. Wählen Sie **Schnellspeichern**, um im Builder zu bleiben, oder **Speichern**, um zu speichern und zurückzukehren.
3. Öffnen Sie das tatsächliche Erfassungs- oder Bearbeitungsformular, das dieses Layout verwendet.
4. Prüfen Sie Feldreihenfolge, Pflichtmarkierungen, Hilfetexte, Nachschlagefelder und Tastaturnavigation.
5. Speichern Sie einen Testdatensatz und prüfen Sie seine Werte.

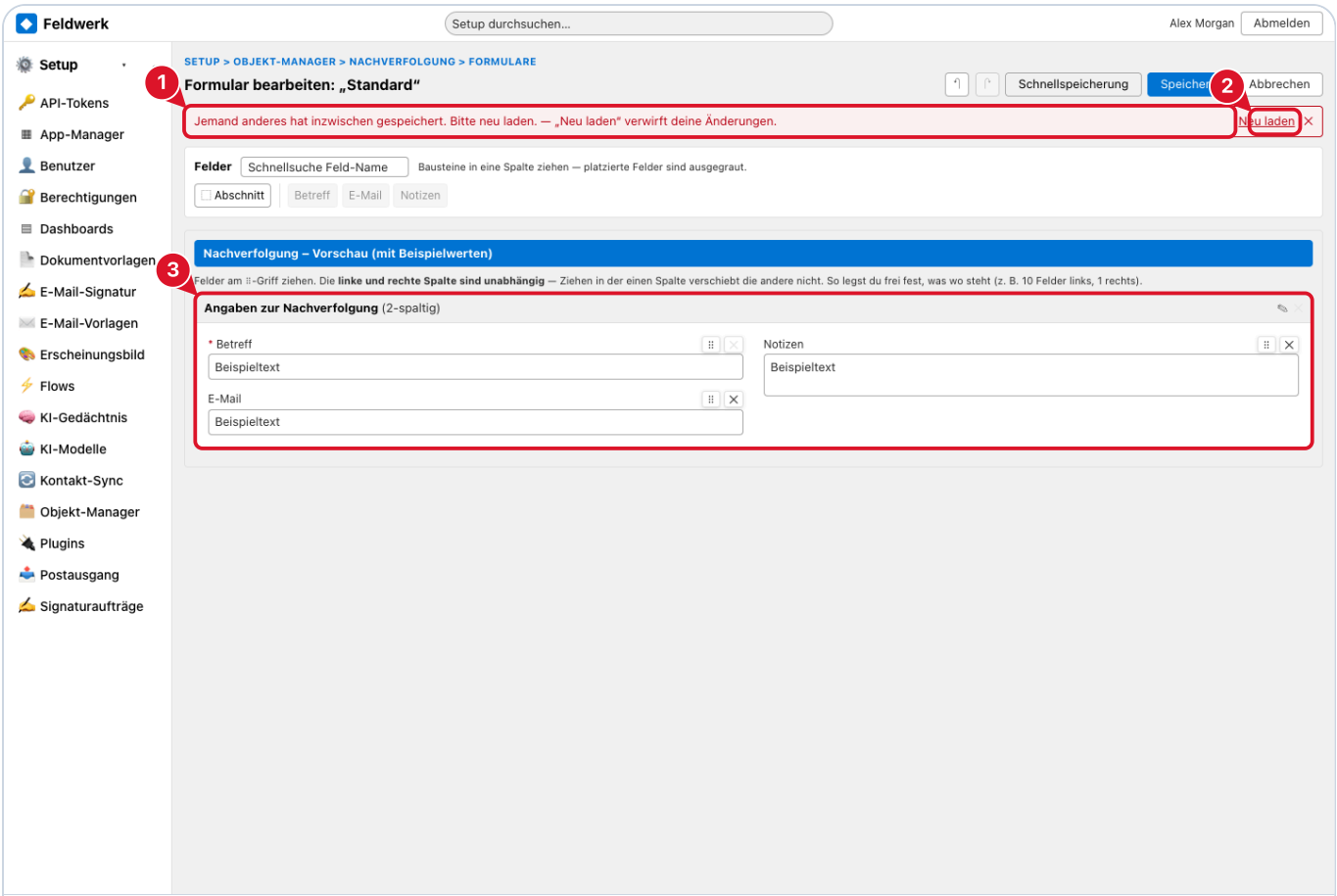
Die Builder-Vorschau verwendet Beispielwerte. Sie beweist nicht, dass ein tatsächlicher Nachschlagewert verfügbar ist oder die Validierung einen Datensatz akzeptiert. Testen Sie sowohl mit den Berechtigungen der vorgesehenen Benutzer als auch mit einem Administratorkonto.

Öffentlich verwendete Formulare

Ein Formular kann nach Aktivierung in der Formularliste auch als öffentliches Webformular dienen. Prüfen Sie vor Veröffentlichung des Links genau die Felder und Texte, die Besucher sehen. Plattformschaltflächen folgen der unterstützten Oberflächensprache; Ihre Beschriftungen und Abschnittsüberschriften bleiben konfigurierte Inhalte.

Wenn Änderungen nicht erscheinen

Prüfen Sie, dass das Speichern erfolgreich war und Sie das tatsächlich im Ablauf verwendete Formular bearbeitet haben. Prüfen Sie mehrere Formulare desselben Objekts und aktualisieren Sie die Zielseite. Meldet das Speichern einen Konflikt durch gleichzeitige Änderungen, hat jemand eine neuere Version gespeichert. Ihr Speicherversuch wird abgelehnt. Wählen Sie **Neu laden**, um das ungespeicherte Layout zu verwerfen und die aktuelle Version zu laden. Wenden Sie danach die weiterhin benötigten Änderungen an. Notieren Sie Ihre beabsichtigten Änderungen vor dem Neuladen. Das Beispiel zeigt ein zweispaltiges Layout, dessen Speicherung abgelehnt wurde; unter dem Hinweis steht weiterhin der lokale Entwurf.



1. Das Speichern wurde abgelehnt, weil eine neuere Version existiert.
2. Neu laden verwirft diesen Entwurf und öffnet das zuletzt gespeicherte Layout.
3. Bis zum Neuladen zeigen diese Felder weiterhin den lokalen Entwurf.

Öffentliche Webformulare

Informationen über ein öffentliches Formular erfassen und den entstandenen Datensatz prüfen.

Was ein öffentliches Webformular ist

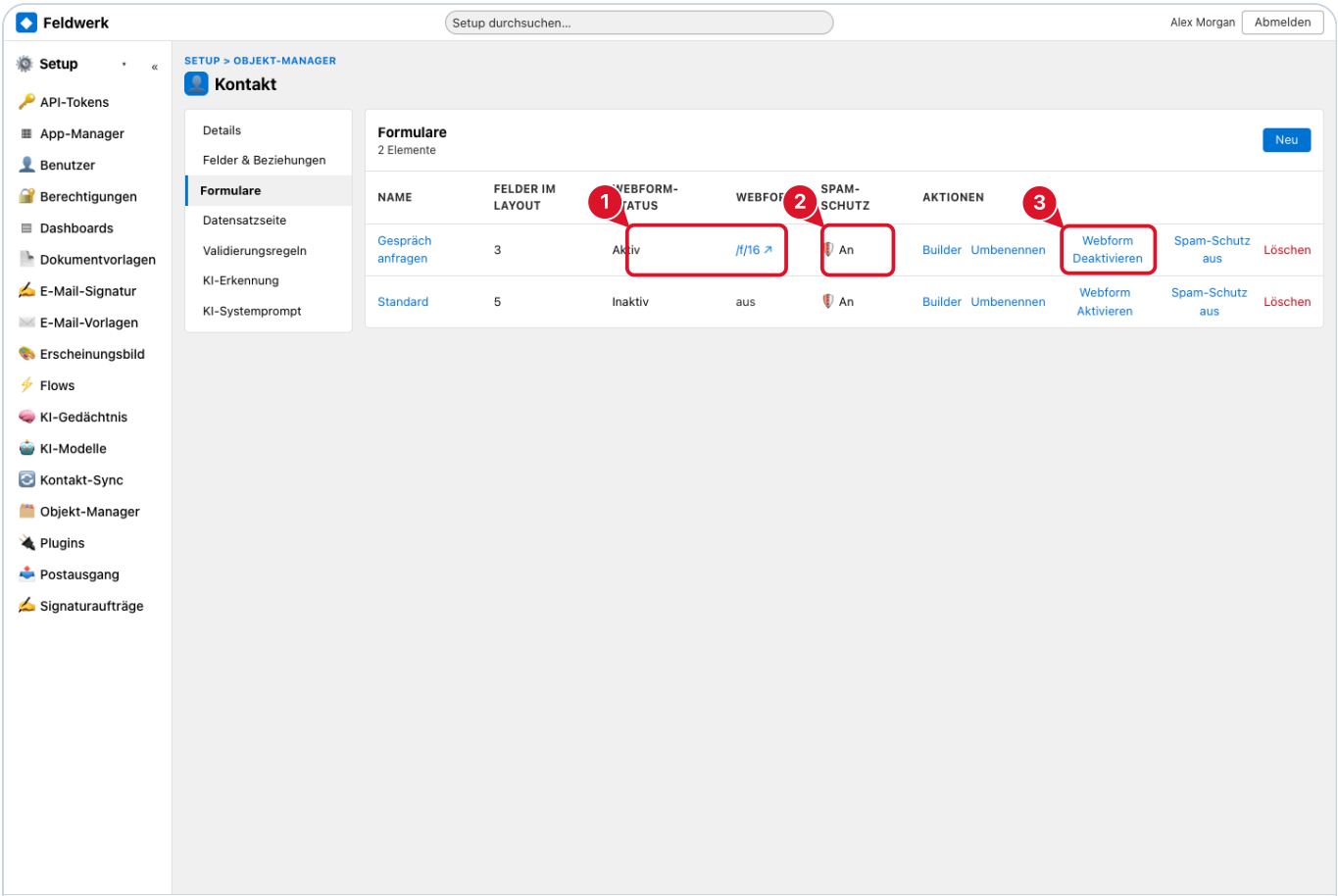
Mit einem öffentlichen Webformular können Personen Informationen übermitteln, ohne sich bei Feldwerk anzumelden. Es basiert auf einem Formular für ein Objekt; die gewählten Felder und das Layout bestimmen die Besucheransicht. Die Übermittlung erstellt einen Datensatz über den normalen Schreibpfad von Feldwerk einschließlich Validierungen und konfigurierter Automatisierungen.

Die Veröffentlichung eines Formulars veröffentlicht nicht die normale Datensatzliste des Objekts. Öffentliche Formulare schließen Nachschlagefelder aus; Besucher können keine verknüpften Datensätze über eine interne Nachschlageauswahl durchsuchen oder auswählen. Wählen Sie nur die Felder, die Sie tatsächlich von Besuchern benötigen. Im Handbuchbeispiel erfasst **Gespräch anfragen** Name, E-Mail-Adresse und Notizen eines Kontakts.

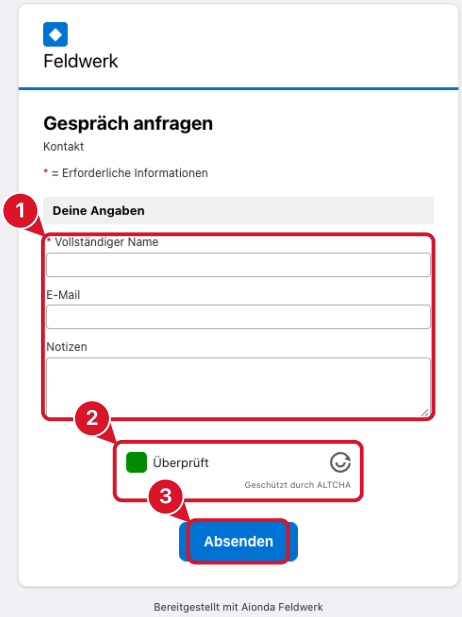
Ein Formular vorbereiten und aktivieren

1. Öffnen Sie als Administrator **Setup → Objekt-Manager** und wählen Sie das Zielobjekt.
2. Öffnen Sie **Formulare**. Erstellen Sie ein eindeutig benanntes Formular oder wählen Sie ein vorhandenes.
3. Öffnen Sie den **Builder** und ordnen Sie Felder und Abschnitte an. Nehmen Sie die Pflichtfelder auf, die Besucher ausfüllen müssen.
4. Speichern Sie das Layout und aktivieren Sie das Webformular über seine Aktion in der Formularliste.
5. Lassen Sie den Spamschutz eingeschaltet. Die Formularliste zeigt Status, öffentlichen Link und Spamschutzzustand des Webformulars.
6. Öffnen Sie den öffentlichen Link in einem separaten Browserkontext ohne angemeldete Feldwerk-Sitzung.

Verwenden Sie den tatsächlich von Feldwerk angezeigten Link, etwa `/f/16` in der Beispielinstanz. Die numerische Formular-ID wird von Ihrer Instanz vergeben und weicht vom Beispiel ab.



1. Aktivstatus und öffentlichen Link prüfen.
2. Spamschutz eingeschaltet lassen.
3. Nach Ende der Erfassung das öffentliche Formular deaktivieren.



1. Erbetene Informationen eingeben.
2. Spamschutzprüfung kontrollieren.
3. Absenden und auf die Bestätigung warten.

Absenden und Ergebnis prüfen

1. Geben Sie für Ihren Test einen fiktiven Namen und eine fiktive E-Mail-Adresse ein.
2. Füllen Sie die mit einem Stern gekennzeichneten Felder aus. Prüfen Sie den Spamschutzstatus und folgen Sie angezeigten Anweisungen.
3. Wählen Sie **Absenden**.
4. Warten Sie auf **Vielen Dank!** und die Bestätigung der Speicherung.
5. Suchen Sie im angemeldeten Arbeitsbereich den neuen Datensatz und prüfen Sie seine Werte sowie erwartete Automatisierungsergebnisse.

Weiteren Eintrag senden öffnet das Formular für eine neue Übermittlung. Dadurch entsteht ein weiterer Datensatz; die vorherige Übermittlung wird nicht bearbeitet. Jede Übermittlung benötigt eine neue Spamschutzprüfung.

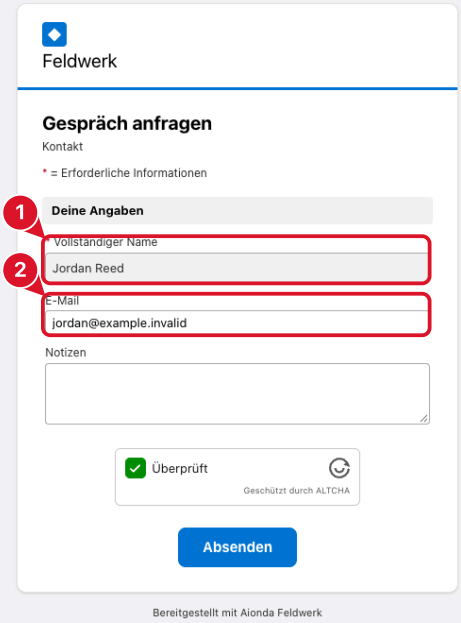
Oberflächensprache und Formularinhalt

Ohne angemeldetes Konto bestimmt die Browsersprache die unterstützte Oberflächensprache. Mit angemeldetem Konto gilt die Kontosprache. Schaltflächen, Statusmeldungen und Spamschutz werden übersetzt. Formulartitel, Abschnittsüberschriften, Feldbeschriftungen und eigene Validierungsmeldungen bleiben wie konfiguriert. Bereiten Sie diese Texte für die Zielgruppe des Formulars vor.

Einen Link vorbelegen

Abfrageparameter können Anfangswerte anhand der Feld-API-Namen liefern. Beispielsweise belegt `/f/16?name=Jordan%20Reed` das Namensfeld vor, sofern es zum Formular gehört. Kodieren Sie Werte für die URL. Ein Parameter `fest` kann ein bereits vorbelegtes Feld im Formular schreibgeschützt darstellen, etwa `&fest=name`.

Diese Parameter sind Bestandteil des Links und können von dessen Empfänger geändert werden. Eine schreibgeschützte Darstellung beweist keine Berechtigung für den übermittelten Wert. Konfigurieren Sie bei referenzbasierten Links die unterstützte serverseitige Referenzprüfung und gegebenenfalls ein alternatives Formular.



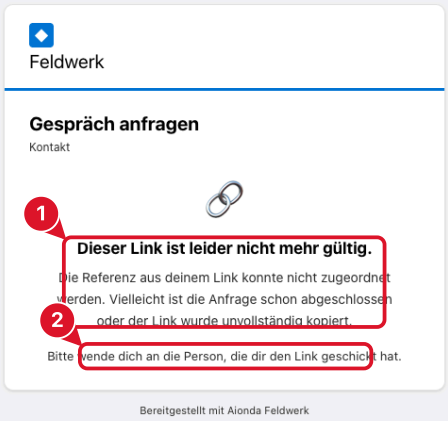
1. Der vorbelegte Name ist schreibgeschützt.
2. Die vorbelegte E-Mail-Adresse bleibt bearbeitbar.

Referenzbasierte Links konfigurieren

Referenzprüfungen sind Feldoptionen, die über die API oder die generischen MCP-API-Werkzeuge konfiguriert werden. Sie sind kein eigener Schalter im Formular-Builder. Setzen Sie am Textfeld für die Referenz `optionen.webformPruefung` mit dem API-Namen des Zielobjekts in `objekt` und dem API-Namen seines Referenzfelds in `feld`. Lesen Sie zunächst das aktuelle Feld und geben Sie beim Aktualisieren seine aktuelle `version` mit. Behalten Sie seine übrigen Optionen bei.

Die übermittelte Referenz muss mit einem vorhandenen Wert in diesem Zielfeld übereinstimmen. Verwenden Sie für personalisierte Links einen eigenen, schwer erratbaren Referenzwert. Ein passender Name oder ein schreibgeschützter URL-Parameter identifiziert den Besucher nicht. Nehmen Sie das Referenzfeld in das öffentliche Formularlayout auf, damit sowohl die anfängliche Linkprüfung als auch die Prüfung beim Absenden greifen.

Optional setzen Sie `fallbackFormularId` auf ein aktives alternatives Formular. Der Ausweichlink übergibt den fehlgeschlagenen Wert als `?referenz=…&fest=referenz`. Nehmen Sie ein geeignetes Feld `referenz` in das alternative Formular auf, wenn es diesen Wert behalten soll. Testen Sie den Ausweichweg vor dem Verteilen der Links. Ohne alternatives Formular bittet Feldwerk die Besucher, die Person zu kontaktieren, die den Link versendet hat.



1. Eine nicht zuordenbare Referenz verhindert das Absenden.
2. Ohne konfiguriertes Alternativformular dem Kontakthinweis folgen.

Häufige Probleme lösen

- **Formular nicht verfügbar:** Prüfen Sie, dass es existiert und als Webformular aktiv ist. Besucher sollten sich an den Absender des Links wenden.
- **Ungültige Referenz:** Der Referenzwert konnte nicht zugeordnet werden. Nutzen Sie ein angebotenes alternatives Formular oder fordern Sie einen neuen Link an.
- **Markierte Felder:** Korrigieren Sie die angegebenen Werte und senden Sie erneut. Ein Datensatz wird nicht allein deshalb angenommen, weil alle sichtbaren Felder Text enthalten.

- **Spamschutz fehlgeschlagen:** Folgen Sie der Meldung und laden Sie das Formular gegebenenfalls neu. Geben Sie ungespeicherte Informationen vor dem nächsten Versuch erneut ein.

Keine weiteren Einsendungen annehmen

Wählen Sie in der Formularliste **Webformular deaktivieren**. Der öffentliche Link zeigt anschließend einen Hinweis auf die Nichtverfügbarkeit. Die Deaktivierung löscht keine bereits erfassten Datensätze; bearbeiten Sie diese im normalen Datensatzablauf.

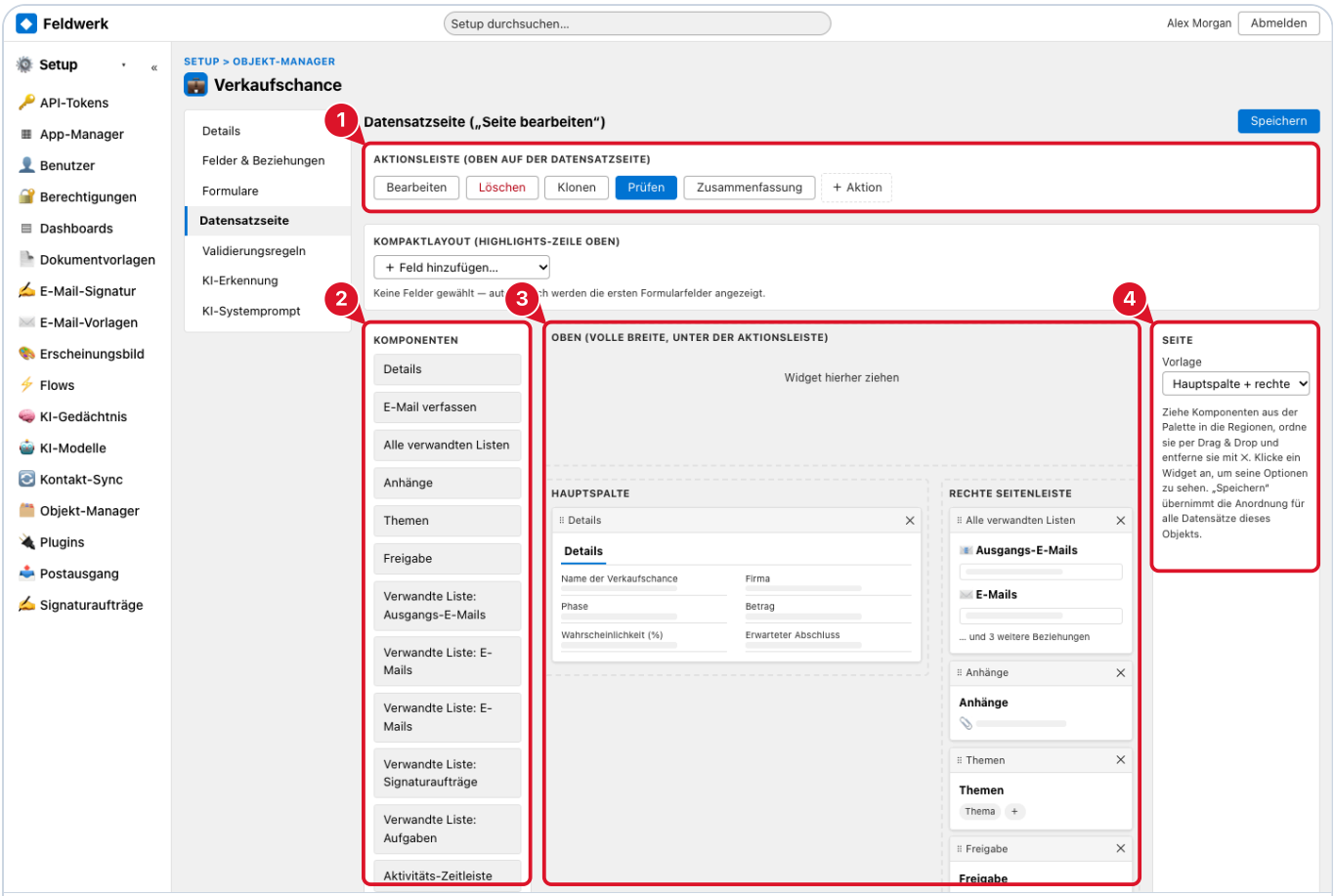
Datensatzseiten-Builder und Aktionen

Die Datensatzseite eines Objekts aus wiederverwendbaren Komponenten und Aktionen zusammensetzen.

Die Datensatzseiten-Konfiguration öffnen

Öffnen Sie als Administrator **Setup → Objekt-Manager → das Objekt → Datensatzseite**. Das Seitenlayout gilt für die Datensätze dieses Objekts. Testen Sie Änderungen mit typischen Datensätzen, auch ohne verwandte Daten, und mit Benutzern mit eingeschränktem Zugriff.

Der Seiten-Builder steuert Aktionsleiste, kompakte Zusammenfassung und Komponentenbereiche. Die Detailkomponente verwendet das Formularlayout des Objekts; ordnen Sie ihre Felder im Formular-Builder neu an.



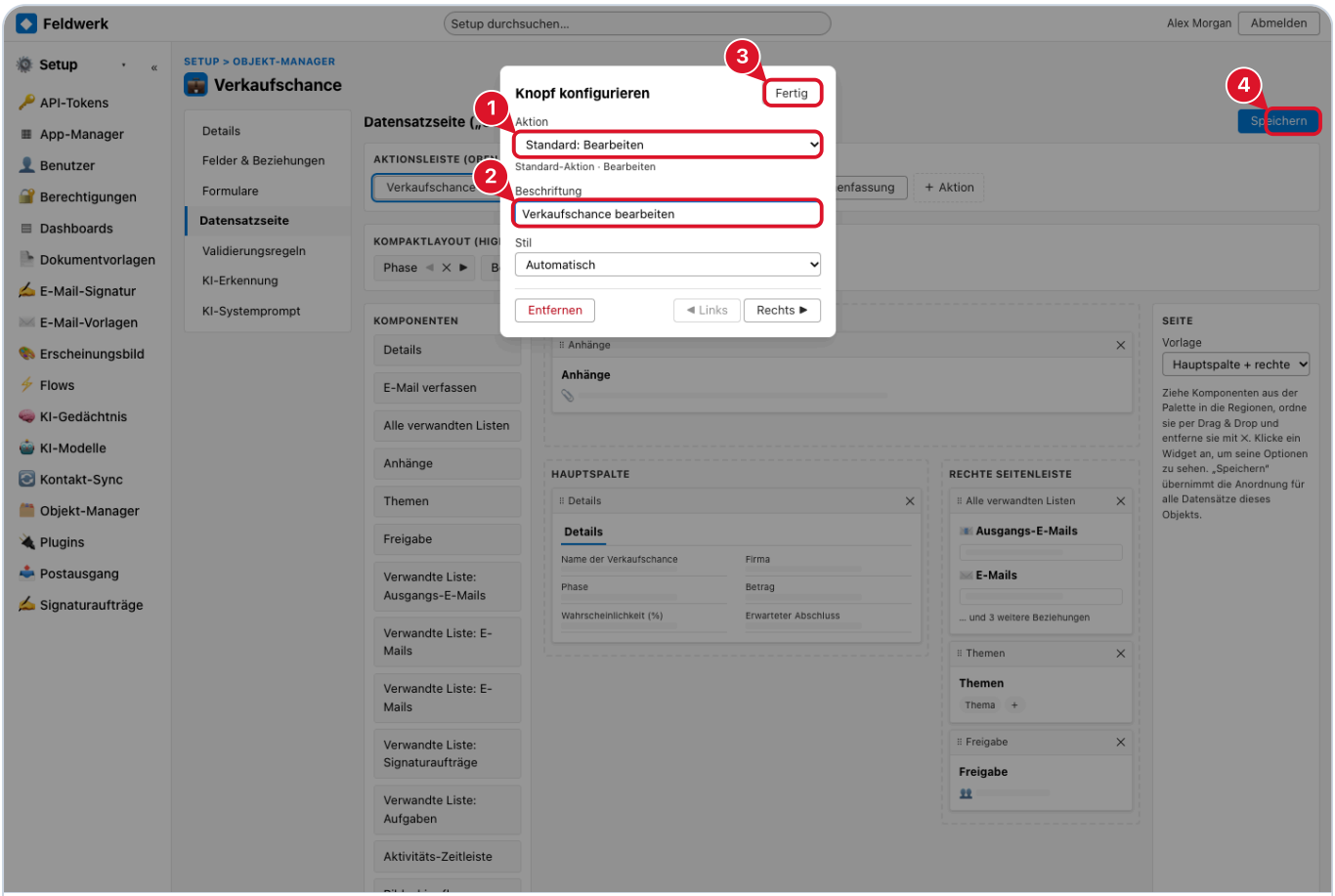
1. Verfügbare Datensatzaktionen konfigurieren.
2. Eine Komponente wählen.
3. Bereiche der Datensatzseite anordnen.
4. Gewählte Komponenten- oder Seitenoptionen prüfen.

Die Aktionsleiste konfigurieren

Fügen Sie mit **+ Aktion** unterstützte Aktionen hinzu. Zu den Standardaktionen gehören Bearbeiten, Löschen und Klonen. Konfigurierte Aktionen können außerdem Bildschirm-Flows starten, Dokumente erzeugen, Plugin-Aktionen aufrufen oder E-Mails verfassen. Wählen Sie eine Beschriftung, die das Ergebnis erklärt, und prüfen Sie die Konfiguration jeder Aktion.

1. Wählen Sie eine vorhandene Schaltfläche in der Vorschau oder **+ Aktion**.
2. Wählen Sie ihre **Aktion**. Beispielsweise öffnet **Standard: Bearbeiten** das Bearbeitungsformular des Datensatzes.
3. Geben Sie eine klare **Beschriftung** ein und wählen Sie einen **Stil**. Die Beschriftung verändert nicht die Wirkung der Aktion.

4. Verschieben Sie die Schaltfläche mit **Links** und **Rechts** oder nehmen Sie sie mit **Entfernen** aus der Leiste.
5. Wählen Sie **Fertig**, um die Konfiguration zu schließen. Die Änderung bleibt zunächst im Builder; erst **Speichern** auf der Seite speichert sie dauerhaft.



1. Die ausgeführte Aktion wählen.
2. Sichtbare Beschriftung festlegen.
3. Fertig schließt den Dialog und behält den Builder-Entwurf.
4. Speichern sichert die vollständige Seitenkonfiguration.

Eine neue Schaltfläche umgeht keine Berechtigungen, Validierungen oder Voraussetzungen. Bearbeiten und direkte Feldbearbeitung benötigen die Bearbeitungsberechtigung am Objekt; Löschen benötigt die Löschberechtigung und Klonen die Erstellungsberechtigung. Diese Standardaktionen werden ausgeblendet, wenn die jeweilige Objektberechtigung fehlt. Bei der Ausführung werden außerdem Datensatzzugriff und Feldberechtigungen geprüft. Testen Sie die Aktion an einem fiktiven Datensatz und prüfen Sie die entstandenen Daten, die Datei oder die eingereihte E-Mail. Eine Wiederholung kann Wirkungen erneut auslösen; verlassen Sie sich nicht allein auf eine Erfolgsmeldung.

Kompakte Zusammenfassung auswählen

Das Kompaktlayout bestimmt die wichtigen Werte in der oberen Zusammenfassungszeile. Wählen Sie mit **+ Feld hinzufügen…** Felder, die Benutzern eine schnelle Identifikation und Beurteilung erlauben. Die Pfeile an jeder Feldmarkierung ändern ihre Position; das Entfernen einer Markierung entfernt das Feld aus der Zusammenfassung. Ohne ausdrücklich gewählte Felder verwendet die Seite automatisch die ersten Formularfelder. Prüfen Sie nach dem Speichern sowohl Beschriftung als auch angezeigten Wert.

Komponenten platzieren

1. Wählen Sie die Seitenvorlage, etwa Hauptspalte + rechte Seitenleiste oder Nur Hauptspalte.
2. Ziehen Sie eine Komponente aus der Palette in den oberen Bereich, die Hauptspalte oder die Seitenleiste.
3. Ordnen Sie vorhandene Komponenten durch Ziehen neu an.
4. Wählen Sie eine Komponente, um ihre Optionen zu konfigurieren.
5. Entfernen Sie unerwünschte Komponenten über deren ausdrückliche Entfernen-Aktion.

Komponente	Zweck
Details	Zeigt das konfigurierte Feldlayout des Datensatzes.
Verwandte Liste / Alle verwandten Listen	Zeigt untergeordnete Datensätze über Nachschlagebeziehungen.
Dateien	Zeigt Anhänge des Datensatzes oder konfigurierter verwandter Quellen.
Themen	Bietet Schlagwörter zum Ordnen von Datensätzen.
Freigabe	Zeigt, soweit erlaubt, Eigentümer und ausdrückliche Freigabesteuerung.
E-Mail verfassen / Aktivitätszeitleiste	Unterstützt konfigurierte Kommunikations- und Aktivitätsabläufe.

Komponente	Zweck
Bildschirm-Flow	Bettet einen mit der Seite verbundenen geführten Ablauf ein.
Pfad	Stellt Phasen eines konfigurierten Auswahllistenfelds dar.
Plugin-Komponente	Zeigt Fähigkeiten eines installierten Plugins.

Beziehungen und Quellen konfigurieren

Wählen Sie für eine verwandte Liste die Nachschlagebeziehung, die auf dieses Objekt verweist. Prüfen Sie die je Zeile angezeigten Felder. Bei Dateien aus verwandten Quellen muss die Bereichsüberschrift die Quelle verständlich benennen, damit Benutzer erkennen, welchem Datensatz ein Anhang gehört.

Ein Pfad muss auf ein geeignetes Auswahllistenfeld verweisen. Ein Bildschirm-Flow muss den vorgesehenen aktiven Flow referenzieren. Derselbe Komponententyp kann sich je nach diesen Einstellungen unterschiedlich verhalten.

Die Seite speichern und prüfen

1. Wählen Sie **Speichern**.
2. Öffnen Sie einen Datensatz dieses Objekts und prüfen Sie Zusammenfassung, Komponentenreihenfolge und Aktionen.
3. Öffnen Sie einen verwandten Datensatz und laden Sie einen harmlosen Testanhang hoch, sofern diese Komponenten zur Seite gehören.
4. Testen Sie eine schmale Bildschirmbreite und einen Benutzer mit dem vorgesehenen Zugriff.

Das Entfernen einer Komponente verändert die Seitenanordnung. Es entfernt keine zugrunde liegenden Datensätze, Anhänge oder Berechtigungen. Schützen Sie Informationen durch Berechtigungen, statt sich auf eine fehlende Komponente zu verlassen.

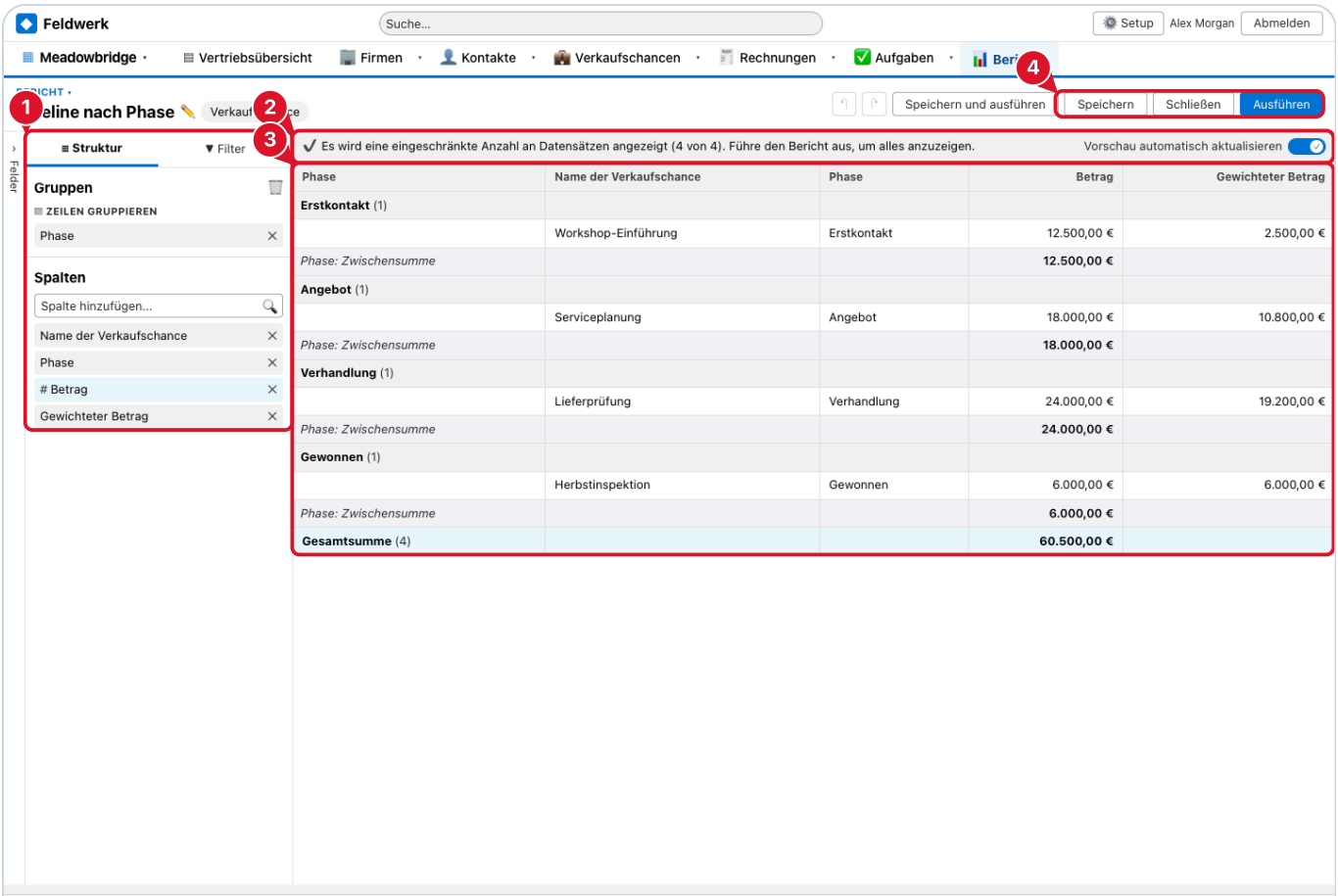
Berichte

Wiederholbare Auswertungen erstellen, das vollständige Ergebnis prüfen und die gewünschten Daten exportieren.

Einen Bericht erstellen

Öffnen Sie **Berichte** in einer App. Wählen Sie **Neuer Bericht**, wählen Sie das Quellobjekt und geben Sie einen aussagekräftigen Namen ein. Vorhandene Berichte lassen sich direkt ausführen oder im Builder öffnen.

Ein Bericht liest Datensätze; er erstellt keinen separaten Datenspeicher. Ergebnisse hängen von Definition, aktuellen Daten und den Zugriffsrechten der betrachtenden Person ab.



1. Spalten, Gruppen und Filter konfigurieren.
2. Umfang der Vorschau prüfen.
3. Gruppierte Werte und Summen kontrollieren.
4. Bericht speichern und ausführen.

Spalten und Gruppen wählen

1. Fügen Sie Felder über den Feldbereich oder die Spaltensuche hinzu.
2. Entfernen Sie unnötige Spaltenmarkierungen und ordnen Sie relevante Spalten an.
3. Fügen Sie eine Zeilengruppierung hinzu, um Datensätze nach einer Kategorie wie Phase zusammenzufassen.
4. Verwenden Sie Spaltenaktionen zum Sortieren, Gruppieren nach einem Feld oder Zusammenfassen eines kompatiblen Werts.

Die Gruppenauswahl im Builder wählt ein einzelnes Gruppierungsfeld. Besitzt der Bericht bereits mehrere über API/MCP konfigurierte Gruppierungsebenen, zeigt **Gruppen** deren Reihenfolge, beispielsweise **Phase → Firma**. **Gruppierung entfernen** löscht alle diese

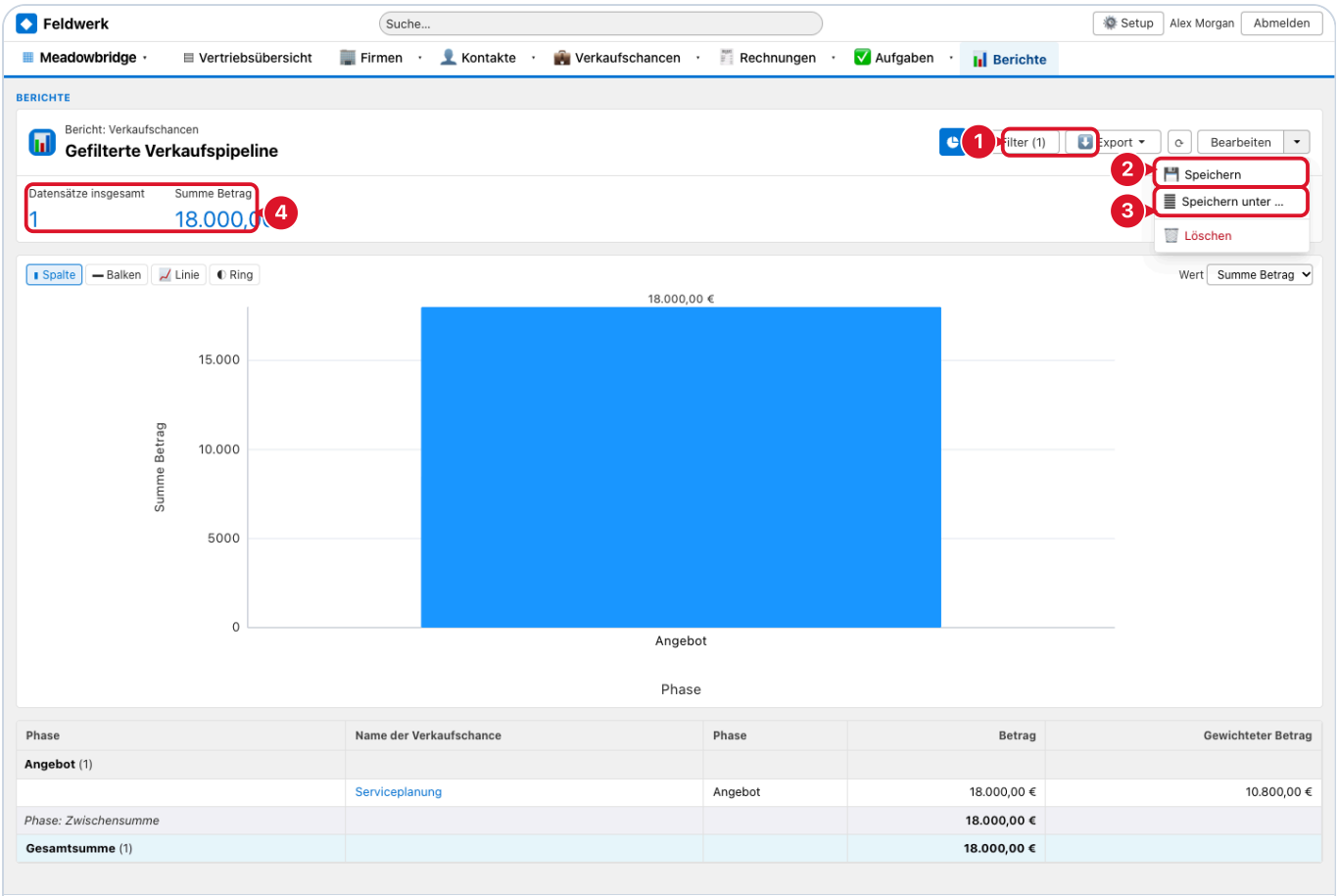
Ebenen. Die Wahl eines neuen Gruppierungsfelds, auch über eine Spaltenaktion, ersetzt die vorhandene Hierarchie durch dieses Feld. Prüfen Sie Änderungen bei Bedarf vor dem Speichern mit **Rückgängig** und **Wiederholen**.

Unser Bericht Pipeline nach Phase gruppiert Verkaufschancen nach Phase und summiert den Betrag. Vier Beispieldatensätze ergeben insgesamt 60500. Gewichteter Betrag ist als Detailspalte enthalten; seine Anzeige ergänzt nicht automatisch eine entsprechende Summenberechnung.

Den Berichtsumfang festlegen

Öffnen Sie Filter, um Feldbedingungen hinzuzufügen. Wählen Sie Operator und Wert bewusst, insbesondere bei Leerwerten, Datumswerten und verknüpften Datensätzen. Relative Datumswerte eignen sich für einen Bericht, der mit dem aktuellen Tag mitwandern soll; ein festes Datum bleibt unverändert.

Wählen Sie im ausgeführten Bericht **Filter**, ergänzen oder bearbeiten Sie eine Bedingung, wählen Sie für diese **Fertig** und anschließend **Anwenden**. Damit wird die aktuelle Auswahl ausgeführt, ohne die gespeicherte Berichtsdefinition zu ändern. Öffnen Sie **Weitere Aktionen**, den Pfeil neben **Bearbeiten**, und wählen Sie **Speichern**, um die angewendeten Filter dauerhaft zu speichern. Mit **Speichern unter …** erstellen Sie stattdessen einen weiteren Bericht mit aktueller Definition und Filtern: Geben Sie den **Namen der Kopie** ein und speichern Sie. Der ursprüngliche Bericht bleibt unverändert. Diese Bearbeitungsaktionen stehen Berichtseigentümern und Administratoren zur Verfügung. Prüfen Sie Name und Umfang, bevor Sie eine gemeinsam verwendete Definition ersetzen.



1. Anzahl angewendeter Filter prüfen.
2. Speichern ersetzt die gespeicherten Filter.
3. Speichern unter erstellt einen separaten Bericht mit diesem Umfang.
4. Vor dem Export Datensatzanzahl und Summe prüfen.

Vorschau und vollständige Ausführung

Die Builder-Vorschau zeigt möglicherweise nur eine begrenzte Datensatzanzahl. Ihr Hinweis nennt den geladenen Umfang. Automatische Vorschauaktualisierungen lassen sich abschalten; aktualisieren Sie die Vorschau dann selbst, um Ihre letzten Änderungen zu prüfen.

Mit **Speichern und ausführen** speichern und starten Sie die Definition. Alternativ verwenden Sie die verfügbaren getrennten Aktionen Speichern und Ausführen. Die Ergebnisansicht zeigt Datensatzanzahl und konfigurierte Kennzahlen. Prüfen Sie Summen anhand bekannter Datensätze, bevor Sie sich auf den Bericht verlassen.

Gruppen, Summen und Diagramme lesen

Gruppenzeilen benennen Kategorie und Anzahl. Zwischen- und Gesamtsummen erscheinen für die konfigurierten Aggregationen. Anzeigeschalter steuern, soweit unterstützt, die Sichtbarkeit von Details oder Summen. Das Ausblenden einer Zeile ändert weder deren Daten noch Berechtigungen.

Mit Diagramm anzeigen prüfen Sie das konfigurierte Diagramm. Ein Diagramm fasst das Berichtsergebnis zusammen und muss im Zusammenhang mit Gruppierung, Aggregation und Filtern gelesen werden. Öffnen Sie den zugrunde liegenden Bericht, wenn ein Dashboarddiagramm Erklärung benötigt.

CSV oder Excel exportieren

1. Führen Sie den Bericht mit den gewünschten Filtern aus.
2. Öffnen Sie **Exportieren**.
3. Wählen Sie, soweit angeboten, **Als CSV (.csv)** oder **Als Excel (.xlsx)**.
4. Öffnen Sie die heruntergeladene Datei und prüfen Sie Überschriften, typische Datensätze und Summen.

Bei einem Bericht mit Zeilengruppierungen enthält der Export Detailzeilen und Gruppierungsspalten. Die auf dem Bildschirm angezeigten Zwischen- und Gesamtsummenzeilen werden diesem Detailexport nicht angehängt. Beispielsweise erzeugt der Filter auf Serviceplanung in dieser Pipeline eine Detailzeile mit Betrag 18000. CSV enthält den angezeigten Währungstext; Excel speichert den Betrag als numerischen Wert mit Währungsformatierung, damit er summiert werden kann. Vergleichen Sie die Detailbeträge mit der Berichtssumme. Ein Export ist eine Momentaufnahme als Datei und aktualisiert sich bei Änderungen der Quelldatensätze nicht. Prüfen Sie seinen Inhalt vor einer Weitergabe außerhalb Ihrer Organisation.

Jahresverlauf, Budgets und Hochrechnungen

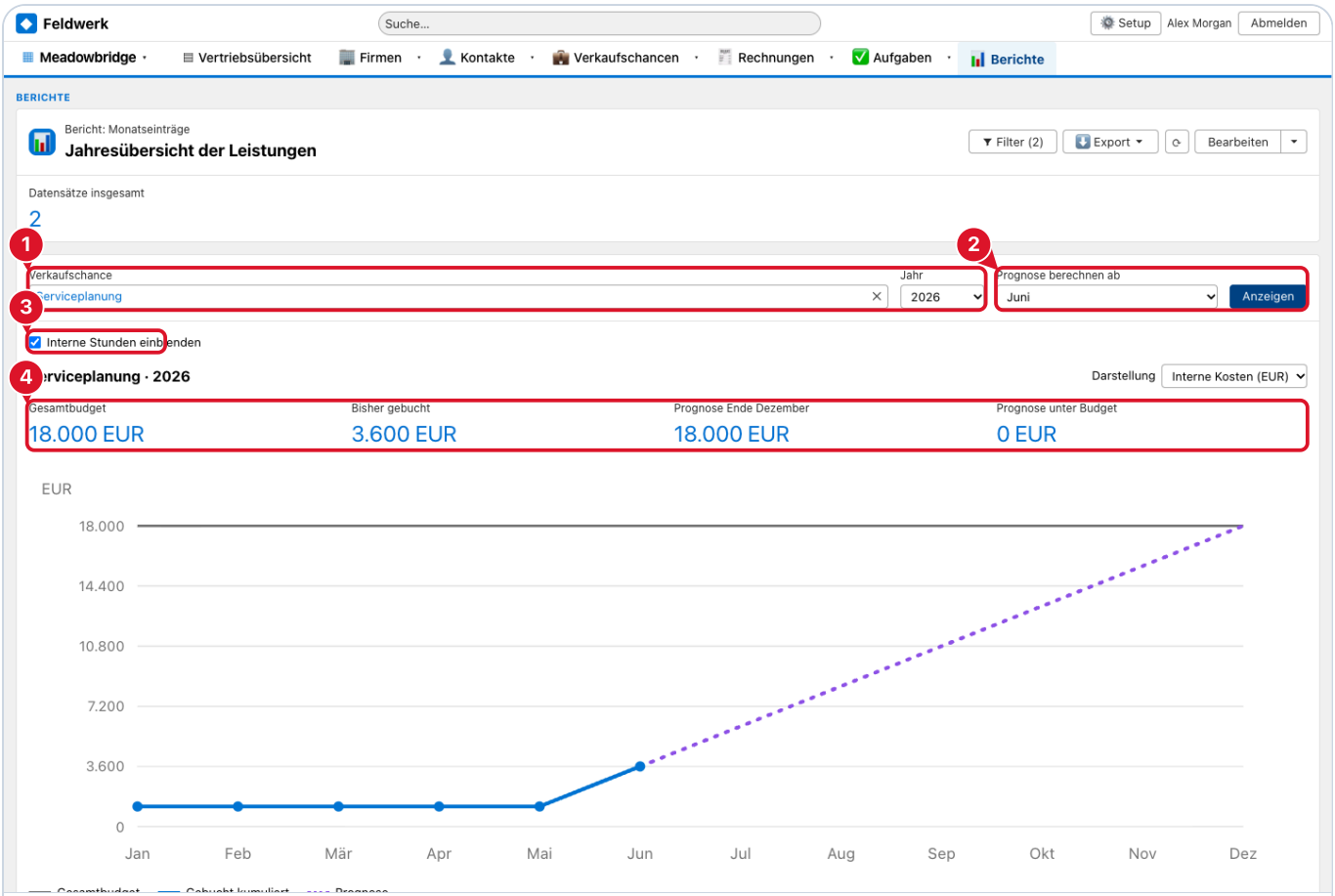
Ein Bericht mit Jahresverlaufskonfiguration ergänzt eine Auswertung von Januar bis Dezember. Die Konfiguration benennt Beziehung, Jahres- und Datumsfelder, Kennzahlen und optionale Budgetfelder. Sie wird über die reguläre API/MCP-Definition eingerichtet; der normale Builder erhält sie.

Wählen Sie Bezugsdatensatz und **Jahr** im ausgeführten Bericht und anschließend **Anzeigen**. Für das aktuelle Jahr ist der Datenstand der letzte Monat mit einer Buchung, nicht automatisch der heutige Monat. Monate bis zu diesem Stand ohne Buchungen zählen als null; spätere Monate bleiben offen. Ein vergangenes Jahr gilt bis Dezember als vollständig. Prüfen Sie den Datenstand vor der Interpretation einer Hochrechnung: Fehlende aktuelle Buchungen können ihn früher als erwartet belassen. Das Budget ist das aktuelle Gesamtbudget des verknüpften Datensatzes, kein historischer Jahresschnappschuss. Ein fehlendes Budget ist unbekannt und unterscheidet sich von einer ausdrücklich gesetzten Null.

Unter **Hochrechnung ab** können Sie automatisch bei der ersten von null verschiedenen Aktivität der ersten konfigurierten Kennzahl beginnen oder einen Monat wählen und mit **Anzeigen** anwenden. Die Auswahl ist vorübergehend, bis Sie sie mit **Weitere Aktionen → Speichern** oder **Speichern unter …** sichern. Zur Summe aller bisherigen Ist-Buchungen wird der Durchschnitt des Basiszeitraums multipliziert mit den verbleibenden Monaten addiert. Nullmonate innerhalb des Basiszeitraums zählen mit; ein laufender Monat wird nach verstrichenen Kalendertagen gewichtet. Angezeigte Methode und Datenstand erklären die Berechnung. Eine lineare Hochrechnung ist kein Personaleinsatzplan.

Unter **Darstellung** wählen Sie die gezeichnete Kennzahl. Optionale Kennzahlen haben ein Kästchen **… anzeigen**; dieses ändert die interne Darstellung, nicht die Zugriffsrechte. Ein konfigurierter Kundenexport wählt genau eine Kennzahl und schließt unabhängig von diesen Sichtbarkeitsschaltern die übrigen internen Kennzahlen, Budgets und Detaildatensätze aus. Seine Menüeinträge lauten **Kundenbericht als CSV** und **Kundenbericht mit Diagramm (.xlsx)**. Die aktuelle Kundenberichtsvorlage verwendet deutsche Überschriften und Erläuterungen, auch bei englischer Anwendungsoberfläche. Die Excel-Datei enthält ein bearbeitbares Liniendiagramm, keinen Screenshot. Prüfen Sie vor dem Herunterladen von Kundenunterlagen den konfigurierten Exportmodus und die Kennzahl.

Im fiktiven Jahresbeispiel enthält Januar 5 Personentage und 1200 EUR interne Kosten, Juni 10 Personentage und 2400 EUR. Mit Juni als Hochrechnungsbeginn und letztem Buchungsmonat bleiben die gesamten bereits gebuchten 3600 EUR erhalten. Hinzu kommen sechs Monate zum Juni-Satz von 2400 EUR; die Hochrechnung beträgt 18000 EUR. Der Kundenexport enthält nur Personentage, auch wenn interne Kosten und interne Stunden intern sichtbar sind. Eine Begrenzung der Detailzeilen auf eine Zeile kürzt diese Jahresberechnung nicht.



1. Bezugsdatensatz und Jahr wählen.
2. Hochrechnungsbeginn wählen und mit Anzeigen anwenden.
3. Optionale Kennzahlen ändern die interne Darstellung.
4. Gebuchte Kosten, Hochrechnung und aktuelles Gesamtbudget vergleichen.

Unerwartete Summen untersuchen

Prüfen Sie Quellobjekt, Filterwerte, Gruppierung, Aggregationsfunktion, Ergebnisgrenzen und Benutzerzugriff. Unterscheiden Sie eine begrenzte Vorschau vom vollständigen Bericht. Prüfen Sie bei Jahresberichten zusätzlich Bezug, Jahr, Budgetquelle und Basiszeitraum der Hochrechnung. Das Speichern einer Berichtsdefinition korrigiert keine falschen Quelldatensätze.

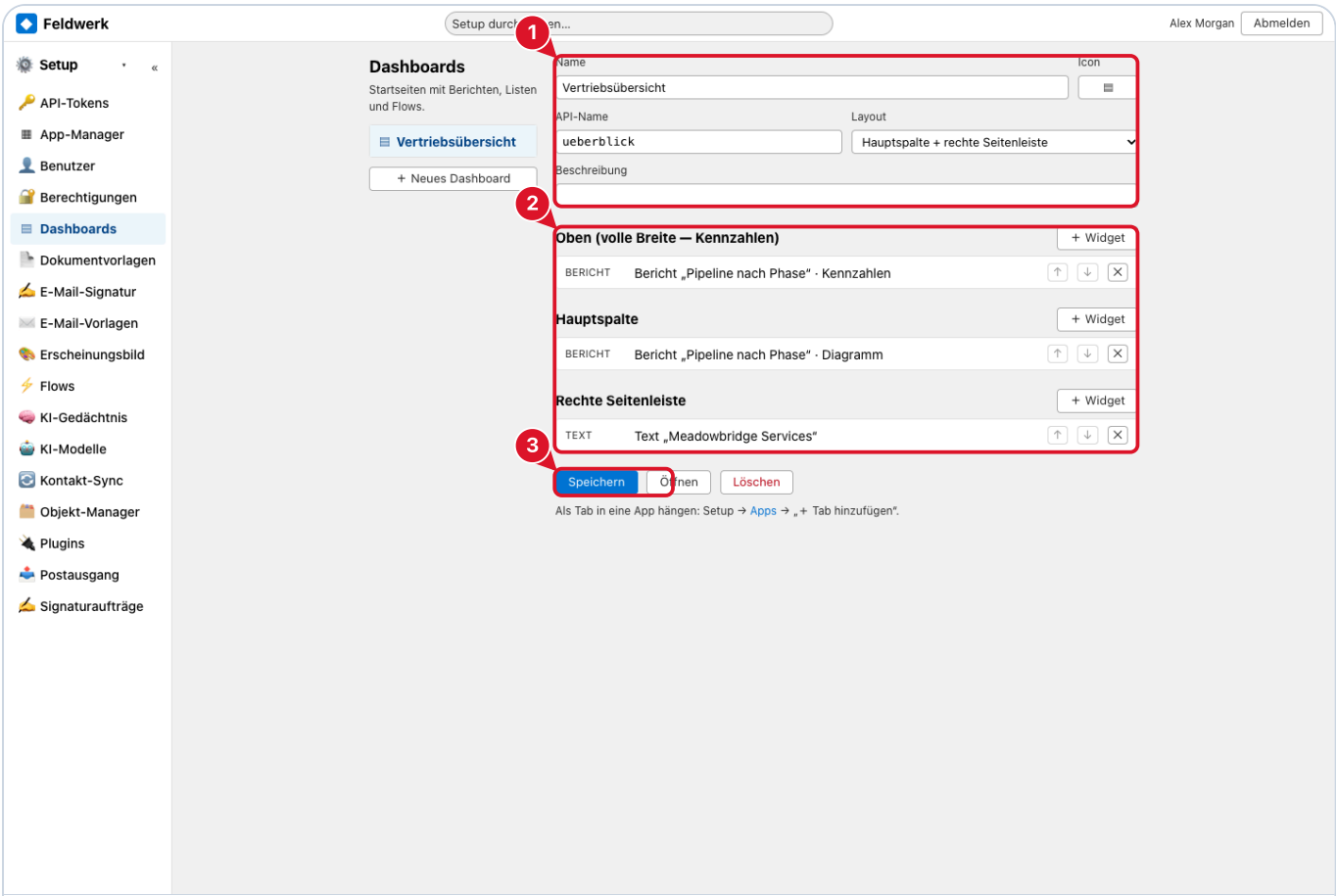
Dashboards

Berichte, Listen, Flows und Hinweise zu einer hilfreichen Startseite kombinieren.

Den Zweck des Dashboards festlegen

Ein Dashboard ist eine Seite ohne einzelnen Datensatzkontext. Es kombiniert wiederverwendbare Berichts-, Listen-, Bildschirm-Flow- und Textkomponenten. Es kann als App-Register hinzugefügt und als Startseite der App verwendet werden.

Die Beispiel-Vertriebsübersicht zeigt Pipeline nach Phase als Kennzahlen und Diagramm sowie einen Textbereich. Ihr Bericht ist derselbe gespeicherte Bericht, der unter Berichte verfügbar ist. Das Dashboard hält keine unabhängige Kopie der Zahlen.



1. Kennung und Layout des Dashboards festlegen.
2. Wiederverwendbare Komponenten hinzufügen und anordnen.
3. Ergebnis speichern und öffnen.

Ein Dashboard anlegen

1. Öffnen Sie als Administrator **Setup → Dashboards**.
2. Wählen Sie **Neues Dashboard**.
3. Geben Sie einen Namen ein, prüfen Sie den API-Namen und wählen Sie Symbol und Beschreibung.
4. Wählen Sie ein Layout: Hauptspalte + rechte Seitenleiste, Zwei gleich breite Spalten oder Eine Spalte.
5. Fügen Sie passende Komponenten im oberen Bereich, im Hauptbereich und in der Seitenleiste hinzu.

Komponenten wählen

Komponente	Konfiguration und Verhalten
Bericht	Gespeicherten Bericht wählen und als Kennzahlen, Diagramm oder Tabelle darstellen. Die Tabellenansicht kann die sichtbaren Zeilen begrenzen.
Liste	Gespeicherte Listenansicht wählen. Ihre Filter und Sortierung bestimmen die Arbeitsliste; das Widget hat eine Grenze für sichtbare Zeilen.
Bildschirm-Flow	Aktiven Flow wählen, der für den Start ohne einzelnen Datensatzkontext ausgelegt ist.
Text	Hinweise, Kontext oder eine Überschrift für die Zielgruppe in Markdown hinzufügen.

Erklären Sie den Umfang mit Komponententiteln, etwa „Offene Verkaufschancen“ statt nur „Gesamt“. Ordnen Sie Komponenten mit den verfügbaren Verschiebeaktionen neu und entfernen Sie solche, die der Aufgabe nicht dienen.

Speichern, öffnen und einer App hinzufügen

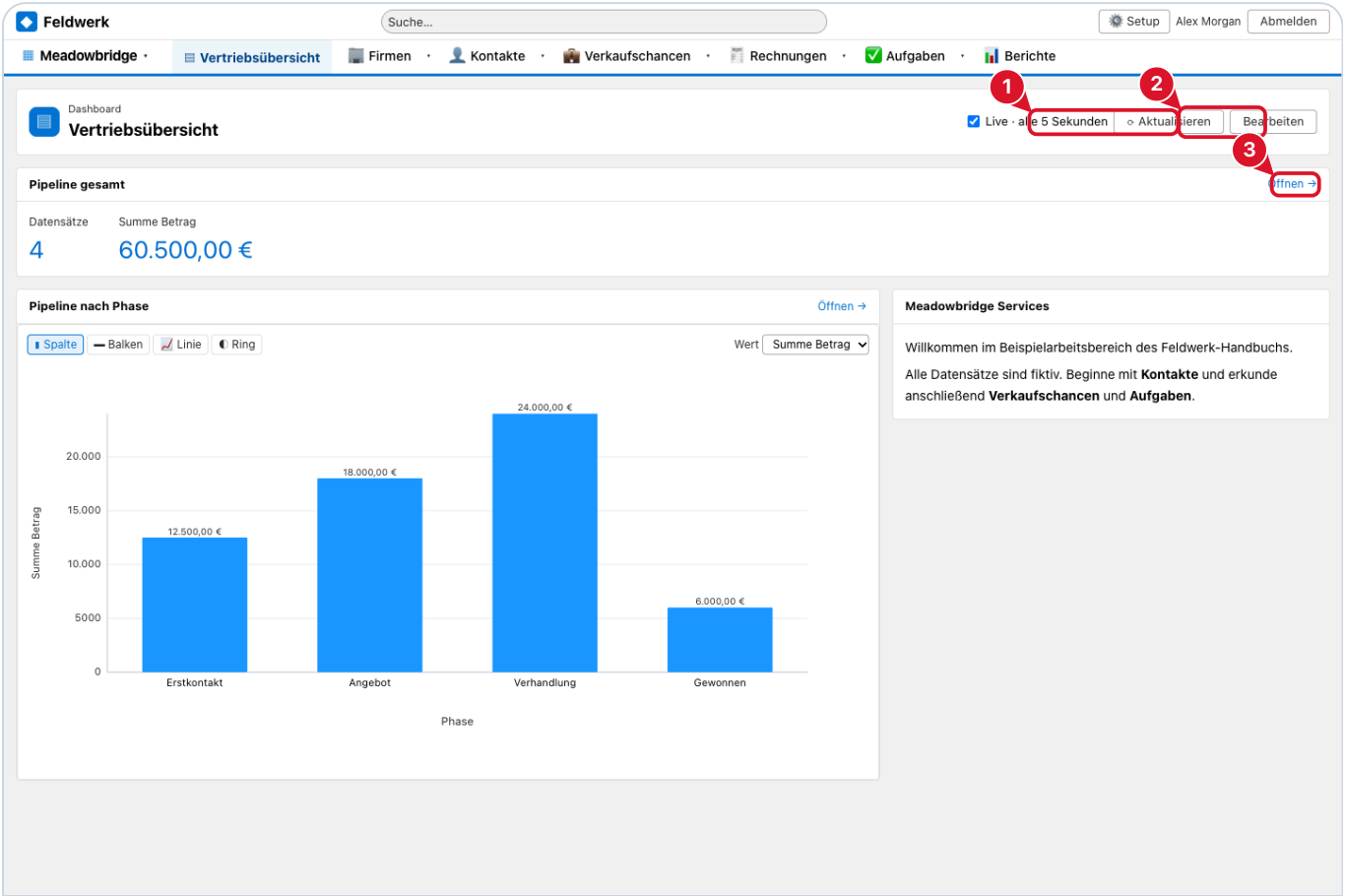
1. Wählen Sie bei einem neuen Dashboard **Dashboard erstellen** oder beim Bearbeiten eines vorhandenen **Speichern**.
2. Wählen Sie **Öffnen** und prüfen Sie die dargestellte Seite.
3. Fügen Sie das Dashboard im App-Manager als Register zur passenden App hinzu.
4. Wählen Sie es als Starteintrag, wenn es die erste Seite dieser App sein soll.
5. Öffnen Sie die App als typischer Benutzer und prüfen Sie das Ergebnis.

Der API-Name bestimmt die Dashboardroute, im Beispiel `/d/ueberblick`. Verwenden Sie den tatsächlich in Ihrer Instanz erstellten Eintrag statt einer Beispielkennung.

Informationen aktualisieren

Mit **Aktualisieren** laden Sie die Datenkomponenten neu. Aktivieren Sie **Live · alle 5 Sekunden** für automatische Aktualisierungen, solange dieser Browser-Tab sichtbar ist. Die Einstellung erscheint als `?live=1` in der Dashboard-URL. Schalten Sie sie aus, um die regelmäßigen Abfragen zu beenden. Die Live-Aktualisierung lädt Datenwidgets neu, ohne einen eingebetteten Bildschirm-Flow neu zu starten.

Mit **Öffnen →** an einer Berichtskomponente öffnen Sie den gespeicherten Bericht und prüfen dessen Filter und vollständige Ergebnisse. Die Zeilengrenze einer Listenkomponente bestimmt nur die im Dashboard angezeigte Anzahl, nicht die gespeicherte Liste oder die Anzahl der Datensätze im Objekt.



1. Regelmäßige Datenaktualisierung bei sichtbarem Tab einschalten.
2. Daten sofort aktualisieren.
3. Quellbericht öffnen, um Filter und Detaildatensätze zu prüfen.

Ein Dashboard erteilt keinen Zugriff

Komponenten laden über die normalen berechtigungsgeprüften Operationen. Benutzer können deshalb auf demselben Dashboard unterschiedliche Ergebnisse sehen. App-Register, Titel und Layout erteilen keine Objekt-, Feld- oder Datensatzrechte.

Eine kleine Tabelle kann nur einen Teil des zugrunde liegenden Ergebnisses zeigen. Öffnen Sie den zugehörigen Bericht oder die Liste für den vollständigen Arbeitskontext. Ein hier eingebetteter Flow muss seine Zieldatensätze selbst erfragen oder finden. Es gibt nicht allein deshalb einen aktuellen Datensatz, weil das Dashboard neben einem Objektregister steht.

Die Seite pflegen

Prüfen Sie Widgets erneut, wenn referenzierte Berichte, Listenansichten oder Flows geändert wurden. Kontrollieren Sie Desktop- und schmale Layouts und halten Sie die wichtigsten Informationen weit oben. Vermeiden Sie dieselbe unerklärte Gesamtsumme in mehreren Komponenten.

Das Löschen eines Dashboards entfernt seine Seite und darauf verweisende Register. Es löscht nicht die von seinen Widgets angezeigten Berichte oder Datensätze. Prüfen Sie die Bestätigung vor dem Löschen.

Wenn ein Widget leer ist oder fehlschlägt

Öffnen Sie den referenzierten Bericht oder die Liste direkt mit demselben Benutzerkonto. Prüfen Sie Filter, Daten und Berechtigungen. Kontrollieren Sie bei einem Flow die Aktivierung und ob er einen Datensatzkontext erwartet. Aktualisieren Sie das Dashboard nach Konfigurationsänderungen und lesen Sie Meldungen, statt ein leeres Widget als Beweis für fehlende Geschäftsdaten zu verstehen.

Mit Flows automatisieren

Automatisierungen mit klaren Auslösern, Bedingungen und überprüfbaren Ergebnissen aufbauen und versionieren.

Den Auslöser vor dem Aufbau wählen

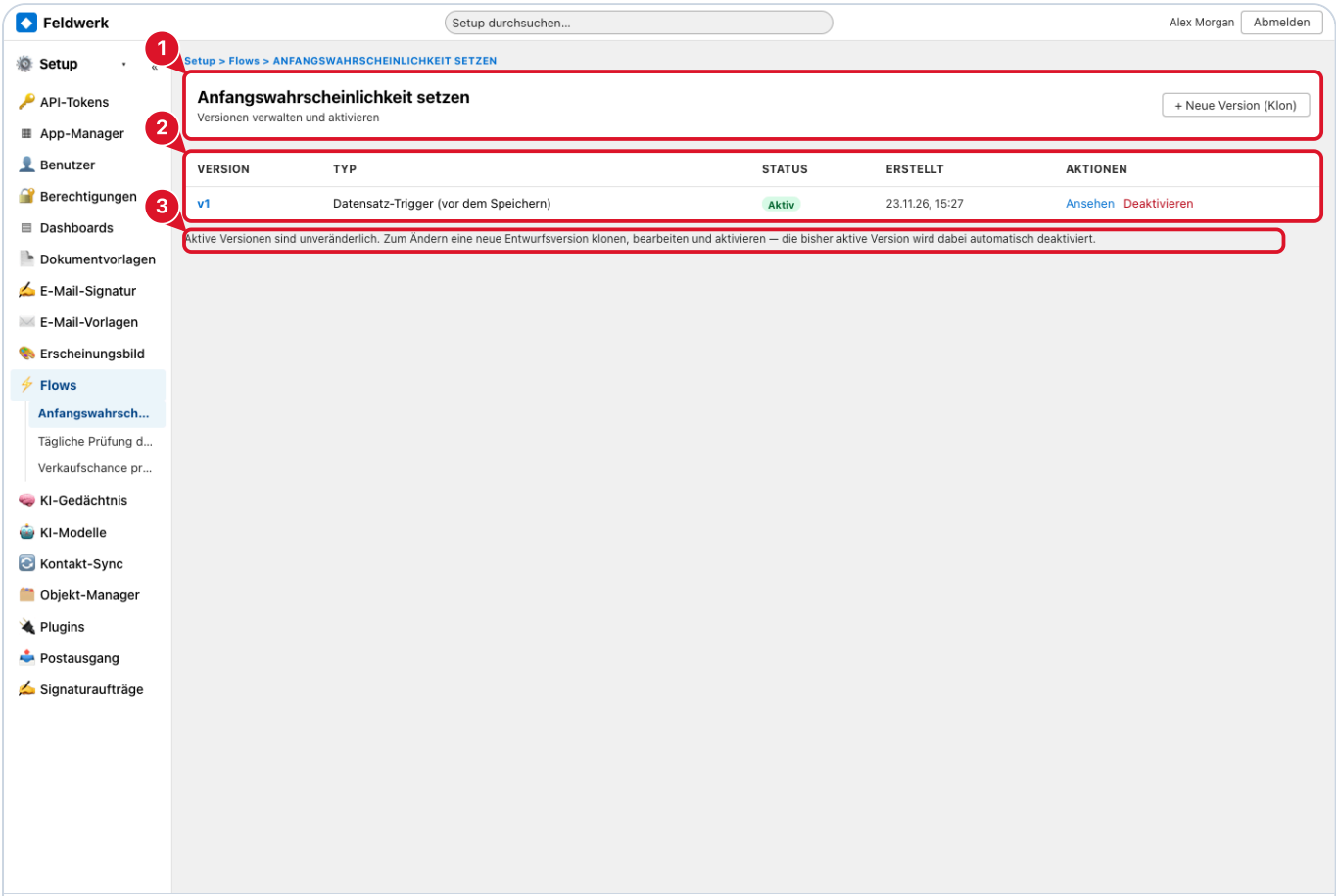
Administratoren verwalten Automatisierungen unter **Setup → Flows**. Eine Flow-Definition besitzt eine stabile Identität und versionierte Logik. Wählen Sie den Auslöser danach, wann die Arbeit stattfinden soll.

Typ	Verwendung
Datensatzgetriggert, vor dem Speichern	Werte am zu speichernden Datensatz vor Validierung und Speicherung setzen.
Datensatzgetriggert, nach dem Speichern	Folgearbeiten nach der Speicherung des Datensatzes ausführen.
Anhang, nach dem Upload	Auf einen neu gespeicherten Datensatzanhang reagieren.
Geplant	Zu konfigurierten Zeiten ausführen, optional für Datensätze, die einen Filter erfüllen.
Bildschirm-Flow	Angemeldete Benutzer durch Eingaben und Aktionen führen.
Eingehende E-Mail	Nachrichten aus einem konfigurierten Postfach mit Eintrittsbedingungen verarbeiten.

Mit Versionen arbeiten

1. Öffnen Sie die Flow-Definition und prüfen Sie ihre Versionen.
2. Erstellen Sie einen neuen Entwurf oder klonen Sie eine neue Version aus einem bestehenden Flow.

3. Bearbeiten und speichern Sie den Entwurf im Flow-Builder.
4. Prüfen Sie Auslöser, Ressourcen, Elemente, Verbindungen und Fehlerpfade.
5. Aktivieren Sie erst nach einem Test des vorgesehenen Verhaltens.



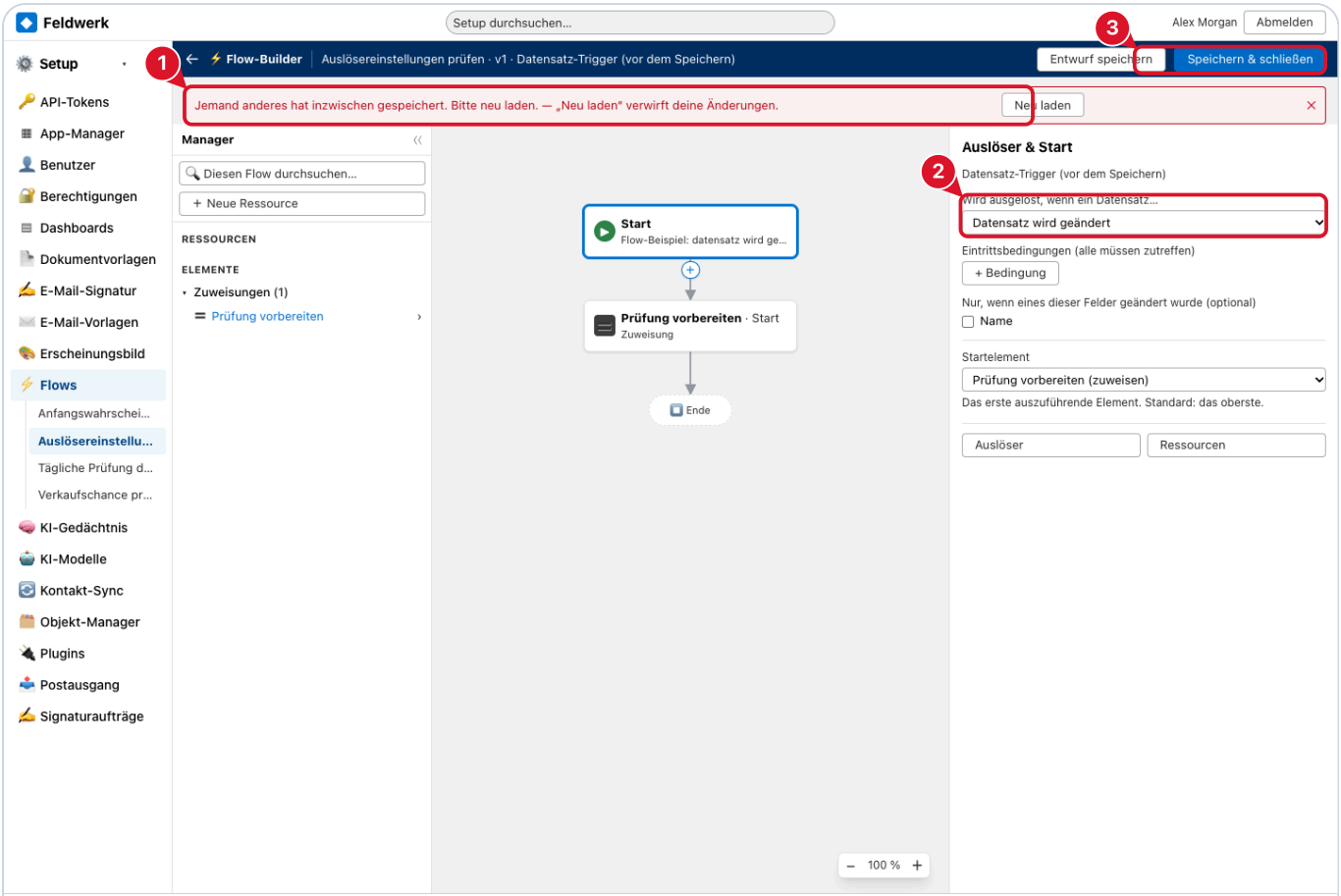
1. Flow-Definition prüfen und einen Ersatzentwurf erstellen.
2. Aktive Version und ihre Aktionen erkennen.
3. Den Ablauf für unveränderliche Versionen beachten.

Aktive Versionen sind unveränderlich. Erstellen Sie zum Ändern einen neuen Entwurf und aktivieren Sie den Ersatz. Die Aktivierung deaktiviert die bisher aktive Version. Laufende Bildschirm-Interviews und Ausführungsprotokolle beziehen sich auf eine bestimmte Version; der Definitionsname allein bezeichnet daher nicht die tatsächlich ausgeführte Logik.

Speichern, ohne den Entwurf zu verlieren

Entwurf speichern lässt den Builder geöffnet. **Speichern und schließen** kehrt erst nach erfolgreichem Speichern zur Versionsliste zurück. Hat jemand dieselbe Version zwischenzeitlich gespeichert, bleiben Ihre lokalen Änderungen sichtbar. Kopieren Sie

benötigte Inhalte vor **Neu laden**: Das Neuladen ersetzt Ihren lokalen Entwurf durch die gespeicherte Version.



1. Konflikthinweis vor Neu laden lesen.
2. Ihre ungespeicherte Auslösereinstellung bleibt sichtbar.
3. Speichern und schließen verlässt den Editor erst nach erfolgreichem Speichern.

Den Ablauf aufbauen

Wählen Sie Start, um Auslöser und Eintrittsbedingungen zu konfigurieren. Ergänzen Sie Elemente für Zuweisungen, Entscheidungen, Schleifen, Datensatzlesen und -schreiben, E-Mail, KI-Abfragen oder unterstützte Plugin-Aktionen. Bildschirmelemente gehören zu interaktiven Flows. Verbinden Sie jeden Zweig mit seinem vorgesehenen Folgeelement und vergeben Sie aussagekräftige Elementnamen.

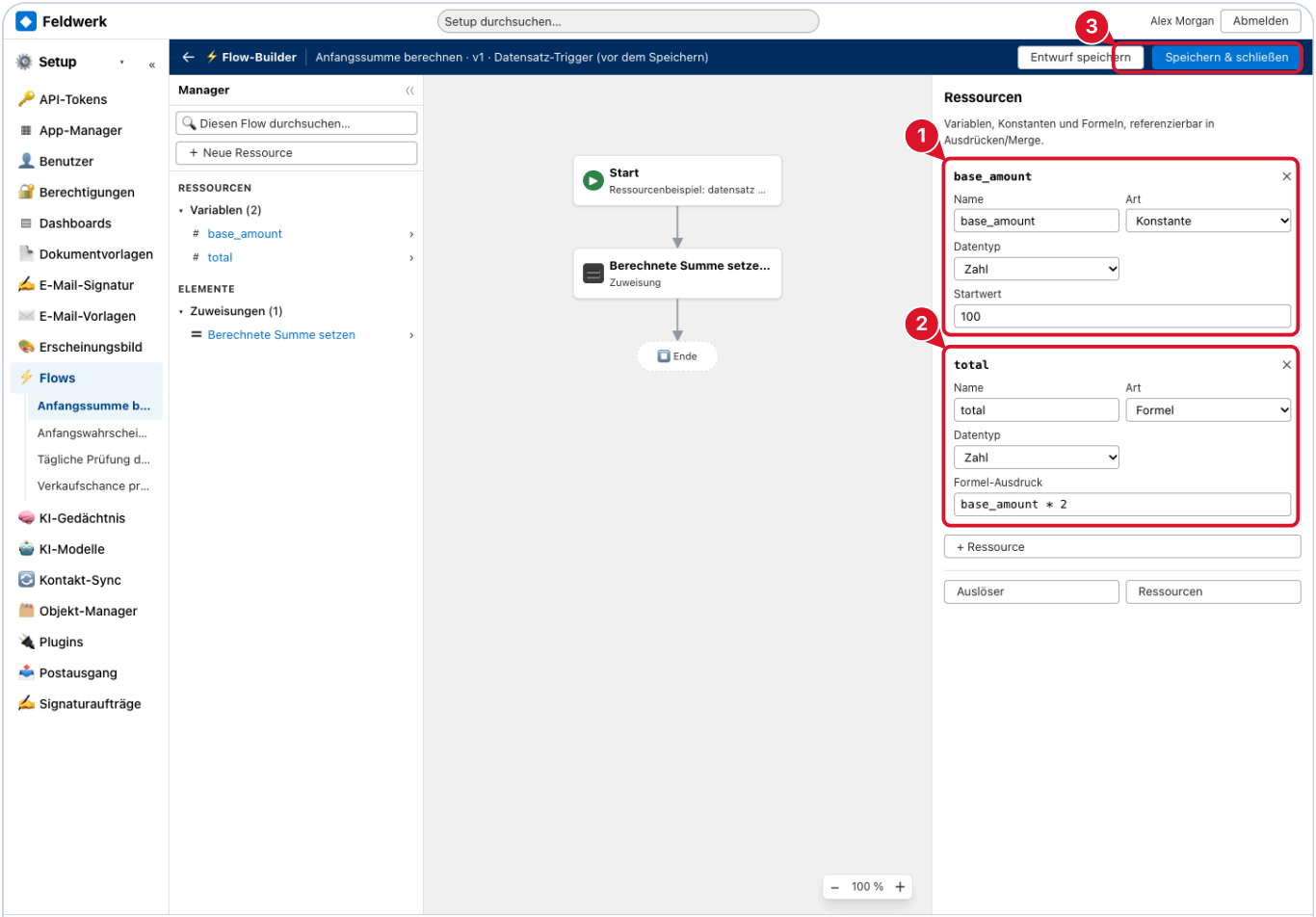
Ressourcen halten Werte, Datensätze oder Sammlungen zwischen Schritten. Wählen Sie den passenden Datentyp und für Datensatz- oder Sammlungsressourcen das zugehörige Objekt. Eine Zuweisung ändert den Flow-Zustand; ein schreibendes Datensatzelement speichert die beabsichtigten Änderungen über den normalen Datensatzdienst.

Ressourcen konfigurieren

Öffnen Sie Ressourcen im Builder und fügen Sie eine Ressource hinzu. Vergeben Sie einen eindeutigen technischen Namen aus Buchstaben, Ziffern und Unterstrichen, beginnend mit einem Buchstaben oder Unterstrich. Ausdrücke beziehen sich auf diesen Namen, etwa `total` oder `current_item.name`.

Art	Verwendung
Variable	Speichert einen Wert aus Zuweisung, Bildschirmeingabe, Datensatzlesen oder Aktion. Skalare und Datensatzvariablen beginnen leer; Sammlungen beginnen als leere Liste.
Konstante	Liefert einen typisierten Startwert. Geben Sie ihn unter Anfangswert ein, etwa 100 für eine Zahl oder true für Ja/Nein. Ein Textstartwert ist wörtlicher Text und kein Ausdruck in Anführungszeichen.
Formel	Geben Sie einen Formelausdruck ein. Er wird während der Ressourceninitialisierung in Ressourcenreihenfolge ausgewertet. Stellen Sie benötigte Anfangswerte davor. Nach späteren Zuweisungen wird er nicht automatisch neu berechnet; verwenden Sie eine weitere Zuweisung an der Stelle, an der neu gerechnet werden muss.
Auswahl	Diese Art erscheint in der Ressourcenauswahl, aber der aktuelle Ressourcenbereich konfiguriert ihre Quelle nicht. Ihre Auswahl allein lädt keine Auswahllistenoptionen. Konfigurieren Sie Bildschirmauswahlen stattdessen in der Bildschirmkomponente.

Der Ressourcenbereich bietet Name, Art, Datentyp, das Objekt für Datensätze/Sammlungen und gegebenenfalls Startwert oder Formel. Eingabe-/Ausgaberichtung und Nullzulässigkeit existieren in der Flow-Definition, haben aber keine separaten Bedienelemente in diesem Bereich. Verwechseln Sie den Datentyp einer Ressource nicht mit der Pflichtfeldeinstellung einer Bildschirmkomponente.



1. Konstante und Anfangswert definieren.
2. Formel nach der Ressource definieren, von der sie abhängt.
3. Entwurf speichern und das gespeicherte Ergebnis vor Einsatz des Flows testen.

Im gezeigten Beispiel beginnt die Zahlenkonstante `base_amount` mit 100; die Zahlenformel `total` wertet `base_amount * 2` aus. Eine Zuweisung vor dem Speichern schreibt `total` in das Quellfeld `gesamt`. Der gespeicherte Testdatensatz enthält 200. Benennen Sie Zwischenressourcen eindeutig, damit sie nicht mit bereits im Flow verfügbaren Quellfeldern kollidieren.

Das passende Element wählen

Element	Konfiguration und Ergebnis
Zuweisung	Eine Ressource setzen, einen numerischen Wert erhöhen oder etwas an eine Sammlung anhängen. Die rechte Seite ist ein Ausdruck; Textliterale wie `"Ready"` benötigen Anführungszeichen. Eine Ressourcenänderung speichert für sich keinen Datenbankdatensatz.
Entscheidung	Geordnete Ausgänge und ihre Bedingungen definieren. Der erste passende Ausgang gewinnt; trifft keiner zu, gilt der Standardausgang. Verbinden Sie erwartete und Ausweichpfade.
Schleife	Sammlung und Ressource für das aktuelle Element wählen. Verbinden Sie den Schleifenkörper zurück mit dem Schleifenelement und den Nach-der-Schleife-Zweig mit dem nächsten Schritt. Ein optionaler Index beginnt bei 1. Eine leere Sammlung führt direkt zum Nach-der-Schleife-Zweig. Das Überschreiten der Iterationsgrenze ist ein Fehler und kein erfolgreicher Abschluss einer Teilliste.
Datensätze abrufen	Objekt, Filter, Sortierung und Ergebnisressource wählen. Erster Datensatz liefert einen Datensatz oder leer; mehrere Datensätze liefern eine Sammlung. Verwenden Sie einen ausdrücklichen Filter und ein zur Aufgabe passendes Limit.

Element	Konfiguration und Ergebnis
Datensätze erstellen	Feldausdrücke angeben und optional den erzeugten Datensatz in einer Datensatzressource speichern. Diese kann anschließend feldweise gelesen oder als Aktualisierungsziel verwendet werden.
Datensätze aktualisieren	Die tatsächliche Datensatzressource oder das konfigurierte Filterziel wählen und Feldausdrücke angeben. Nach der Ausführung die gespeicherten Werte prüfen.
Bildschirm	Eingaben erfassen oder Ergebnisse in einem interaktiven Flow anzeigen. Siehe Bildschirm-Flows für Pflichteingaben, Navigation und Downloads.
Aktion	Eine Aktion aus dem installierten Katalog wählen, Parameter zuordnen und optional das Ergebnis speichern. Den unterstützten Ausdrucks- oder Platzhaltertextmodus des Parameters verwenden. Siehe Plugins und Integrationen.
KI-Abfrage	Einen Prompt einem konfigurierten Modell zuordnen und eine Textantwort oder strukturierte Felder erfassen. Das Fehlerverhalten ausdrücklich wie unten beschrieben einstellen.

Bestehende Flows können das ältere Element E-Mail senden enthalten. Neue E-Mail-Abläufe verwenden die Katalogaktion `email.senden`. Im aktuellen Flow-Graph gibt es kein eigenes Element Datensätze löschen.

KI-Ausgaben in einem Flow verwenden

1. Erstellen Sie Ressourcen für die erwartete Ausgabe: Text für eine freie Antwort oder passende skalare Typen für strukturierte Felder.

2. Fügen Sie eine KI-Abfrage hinzu und wählen Sie ein konfiguriertes Modell oder den Setup-Standard. Ein ausdrücklich ausgewähltes, nicht verfügbares Modell wird nicht stillschweigend durch ein anderes ersetzt.
3. Geben Sie die Aufgabe im Prompt ein, optional mit Systemanweisung. Setzen Sie Datensatzwerte mit Schreibweisen wie `{!$Datensatz.notizen}` ein.
4. Wählen Sie für eine Textantwort die Zielressource. Definieren Sie bei strukturierter Ausgabe jede Zielressource, ihren Ausgabetyp und eine Beschreibung des erwarteten Werts.
5. Legen Sie fest, ob eine nicht verfügbare oder fehlgeschlagene KI mit leeren Ergebnissen fortfahren oder den Fehlerpfad nehmen beziehungsweise stoppen soll. Ergänzen Sie vor jeder Datensatzaktualisierung eine Entscheidung zur Ergebnisprüfung.
6. Speichern Sie geprüfte Werte mit den normalen Elementen Datensätze erstellen oder Datensätze aktualisieren. Die KI-Ausgabe allein speichert kein Feld.

Prompts und darin enthaltene Datensatzwerte werden vom konfigurierten Anbieter verarbeitet. Verwenden Sie nur die für die Aufgabe benötigten Daten. Leere strukturierte Felder sind möglich; ein nach KI-Fehlern fortgesetzter Flow darf bestehende Werte nicht versehentlich löschen. Das Ausführungsprotokoll vermerkt übersprungene KI-Schritte. Zur gemeinsamen Modelleinrichtung siehe KI-Modelle und Konfiguration.

Den richtigen Ausdruckskontext verwenden

Referenzieren Sie aktuelle Datensatzfelder in Flow-Bedingungen als `$Datensatz.feld` mit dem tatsächlichen Feld-API-Namen. Beispielsweise bezeichnet `$Datensatz.wahrscheinlichkeit` das Wahrscheinlichkeitsfeld. Ein Vergleich kann rechts einen festen Wert oder einen unterstützten Ausdruck verwenden.

Zuweisungsziele und Datensatzressourcen sind getrennte Konfigurationsentscheidungen. Im Handbuchbeispiel vor dem Speichern lautet das feldgebundene Zuweisungsziel `wahrscheinlichkeit` und der Ausdruck `20`. Gehen Sie nicht davon aus, dass jede Zieleingabe dieselbe Syntax wie eine Bedingung akzeptiert.

Beispiel: eine Anfangswahrscheinlichkeit setzen

1. Erstellen Sie einen Flow vor dem Speichern für Verkaufschancen, ausgelöst beim Anlegen eines Datensatzes.
2. Ergänzen Sie die Eintrittsbedingung, dass `$Datensatz.wahrscheinlichkeit` leer ist.

3. Fügen Sie eine Zuweisung hinzu, die die Wahrscheinlichkeits-Feldressource auf `20` setzt.
4. Verbinden Sie Start mit der Zuweisung, speichern und aktivieren Sie die Version.
5. Erstellen Sie eine Test-Verkaufschance mit Betrag 500 und ohne Wahrscheinlichkeit.
6. Lesen Sie sie erneut: Wahrscheinlichkeit sollte 20 und Gewichteter Betrag 100 sein.

Testen Sie auch einen Datensatz mit ausdrücklich gesetzter Wahrscheinlichkeit, damit der Standard sie nicht überschreibt. Ein Flow zur Vorbelegung sollte eine enge Bedingung haben und nicht bei jeder späteren Bearbeitung laufen.

Das tatsächliche Schreibziel wählen

Geben Sie für Datensätze aktualisieren die zu aktualisierende Datensatzressource an. Ein häufiges Muster ist Datensätze abrufen in eine Datensatzressource, danach Datensätze aktualisieren mit dieser Ressource. Prüfen Sie anschließend den gespeicherten Wert. Ein erreichter Abschlussbildschirm beweist für sich noch keine Datensatzänderung.

Flows vor dem Speichern aktualisieren die Werte ihres eigenen Datensatzes. Verwenden Sie für Arbeiten an weiteren Datensätzen Flows nach dem Speichern oder einen anderen passenden Flow-Typ. Automatische und geplante Flows können mit Systemrechten laufen; ihre Bedingungen und Ziele erfordern daher eine sorgfältige administrative Prüfung. Bildschirm-Flows verwenden den Zugriff des ausführenden Benutzers.

Beispiel: Nach dem Speichern eine Nachverfolgung anlegen

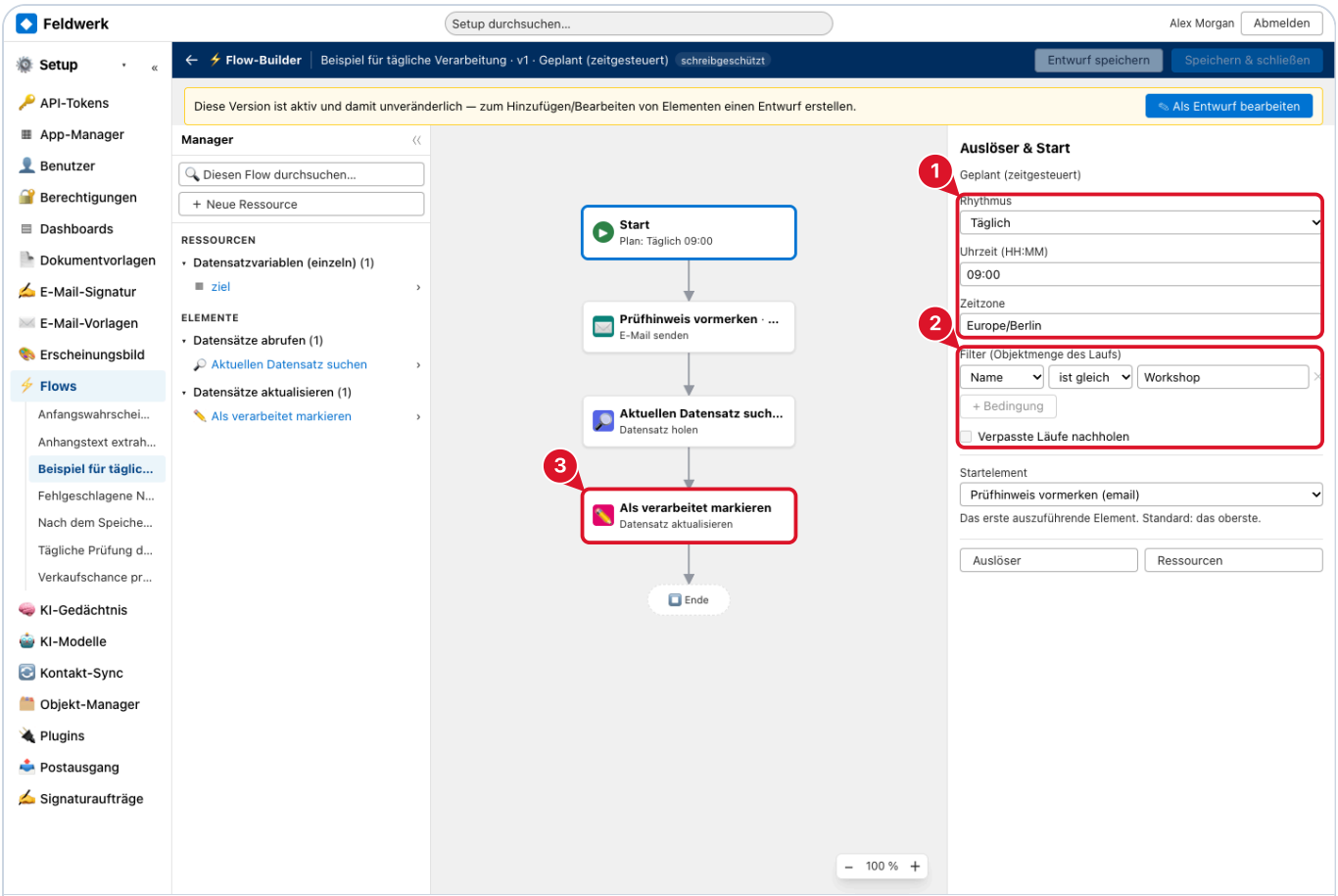
1. Erstellen Sie einen Flow nach dem Speichern für das Quellobjekt. Wählen Sie das Ereignis Erstellt und eine Eintrittsbedingung wie `$Datensatz.ausfuehren` gleich `true`.
2. Fügen Sie Datensätze erstellen für das Nachverfolgungsobjekt hinzu. Belegen Sie alle Pflichtfelder. Ein Namensausdruck wie `"Prüfen " & $Datensatz.name` kombiniert ein Literal mit dem Quellnamen.
3. Verbinden Sie Start, speichern Sie den Entwurf und aktivieren Sie ihn in der Versionsliste.
4. Erstellen Sie einen passenden und einen nicht passenden Quelldatensatz. Nur der passende soll eine Nachverfolgung erzeugen. Lesen Sie beide Objekte erneut.

Im geprüften Beispiel erzeugte Workshop den Eintrag Prüfen Workshop; Nicht bereit erzeugte nichts. Eine Ersatzversion änderte den erzeugten Namen. Ihre Aktivierung deaktivierte die ursprüngliche Version. Deaktivieren Sie die aktive Version, um künftige

automatische Ausführungen zu stoppen.

Beispiel: Datensätze nach Zeitplan verarbeiten

1. Erstellen Sie einen geplanten Flow und öffnen Sie **Auslöser / Start**.
2. Wählen Sie Täglich, geben Sie **09:00** ein und wählen Sie die Zeitzone **Europe/Berlin**.
3. Ergänzen Sie einen Filter für die vorgesehenen Datensätze. Das Beispiel wählt Name gleich Workshop.
4. Lesen Sie den Zieldatensatz in eine Datensatzressource und setzen Sie Verarbeitet mit Datensätze aktualisieren auf true. Ergänzen Sie eine E-Mail-Aktion nur bei benötigter Benachrichtigung.
5. Speichern und aktivieren Sie. Stellen Sie sicher, dass der Server-Scheduler aktiv ist. Prüfen Sie nach dem geplanten Zeitpunkt gespeicherten Datensatz, Ausführungsdatensatz und gegebenenfalls eingereihte E-Mail.



1. Häufigkeit, Ortszeit und Zeitzone festlegen.
2. Datensätze einschränken und Nachholverhalten wählen.
3. Verarbeitungsergebnis mit Datensätze aktualisieren speichern.

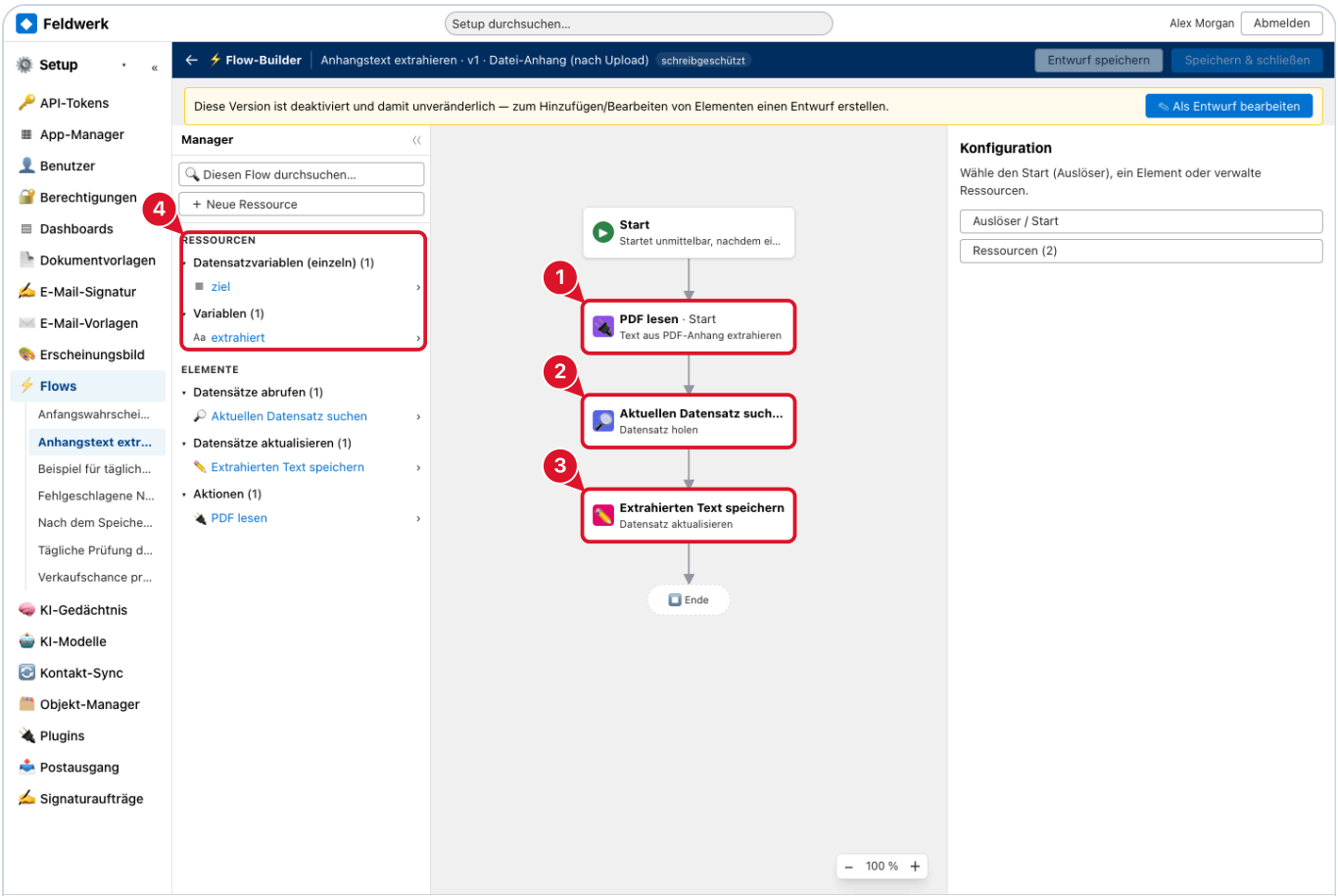
Der geprüfte Zeitplan lief einmal für sein Zeitfenster; ein zweiter Scheduler-Takt erzeugte weder eine doppelte Ausführung noch eine weitere eingereihte Nachricht. Das Einreihen beweist keine Zustellung. Die Abbildung enthält das vorhandene Element E-Mail senden; wählen Sie für neue Flows die unterstützte Aktion `email.senden` aus dem Aktionskatalog.

Tägliche Zeitpläne folgen der konfigurierten Zeitzone einschließlich Sommerzeitwechseln. Stündliche Zeitpläne verwenden stabile Stundenfenster. **Verpasste Läufe nachholen** arbeitet nach einem früheren Lauf verpasste Zeitfenster ab, höchstens 50 pro Prüfung. Ohne Nachholen läuft nur das neueste fällige Zeitfenster. Bei der ersten Prüfung wird ebenfalls nur das neueste berücksichtigt. Deaktiviertes Nachholen bedeutet daher nicht, dass ein verspäteter Scheduler niemals ausführt.

Unterstützte Cron-Muster sind `M H * * *` (täglich), `M * * * *` (stündlich) und `* * * * *` (jede Minute), mit numerischen Minuten- und Stundenwerten an den angegebenen Stellen. Gehen Sie nicht davon aus, dass Bereiche, Listen oder Wochentagsausdrücke unterstützt werden.

Beispiel: Text aus PDF-Anhängen durchsuchbar machen

1. Erstellen Sie am datensatzführenden Objekt einen Flow Dateianhang (nach Upload). Begrenzen Sie die Verarbeitung durch Eintrittsbedingungen.
2. Ergänzen Sie eine Textressource, etwa `extrahiert`, und eine Einzel-Datensatzressource, etwa `ziel`, für dasselbe Objekt.
3. Fügen Sie die Aktion `dokument.pdf_text_extrahieren` hinzu und speichern Sie ihre Ausgabe in `extrahiert`.
4. Fügen Sie Datensätze abrufen für das aktuelle Objekt hinzu. Vergleichen Sie ID mit dem Ausdruck `$Datensatz.id`, lesen Sie den ersten passenden Datensatz und speichern Sie ihn in `ziel`.
5. Fügen Sie Datensätze aktualisieren mit `ziel` als Datensatzressource hinzu. Ordnen Sie dem Ziel-Langtextfeld den Ausdruck `extrahiert` zu.
6. Verbinden Sie die Elemente, speichern und aktivieren Sie. Laden Sie eine PDF mit auswählbarem Text hoch, lesen Sie den Datensatz erneut und suchen Sie eine charakteristische Formulierung.



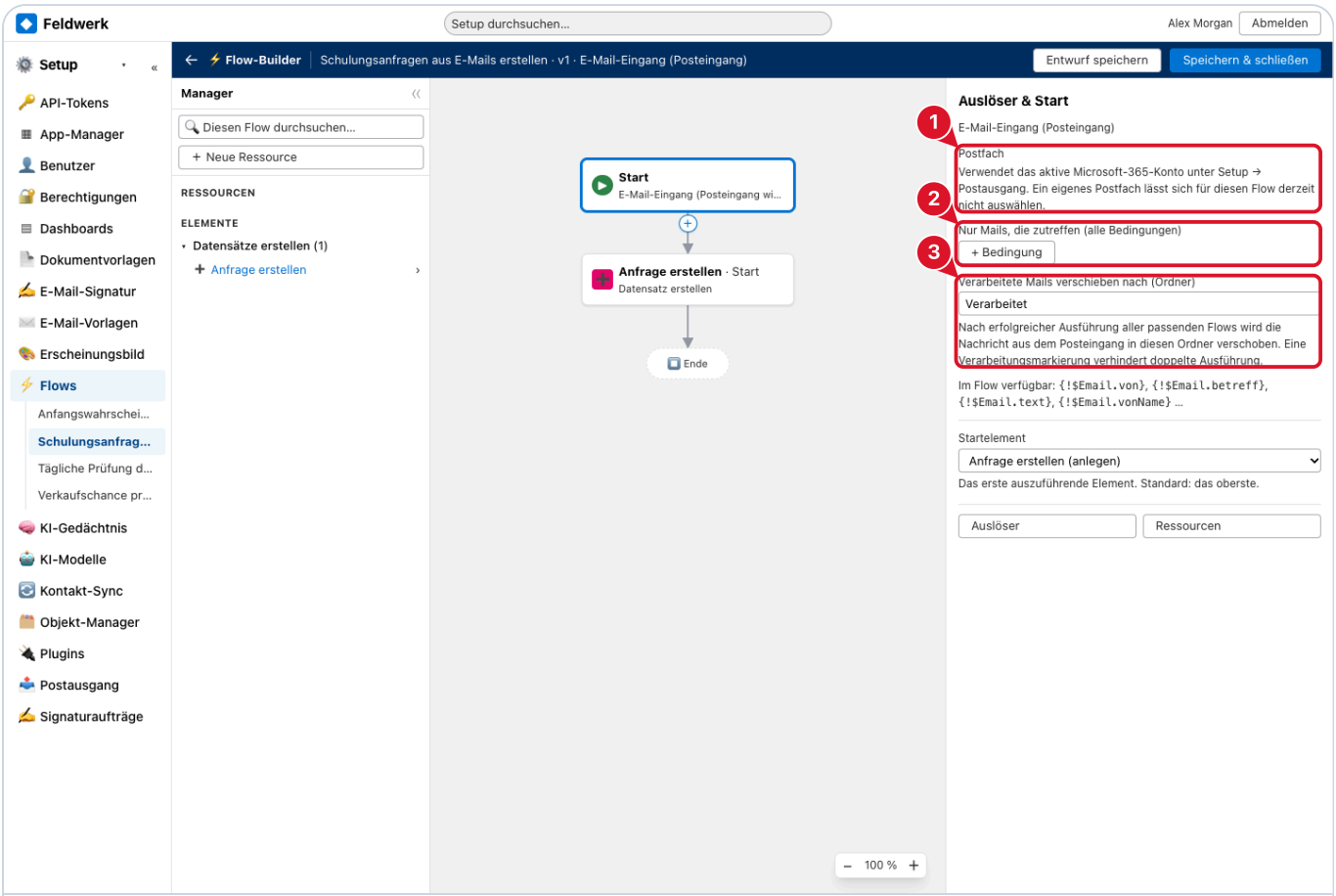
1. PDF-Text lokal extrahieren.
2. Aktuellen Datensatz in die Zielressource lesen.
3. Extrahierten Text im Zielfeld speichern.
4. Textergebnis und Zieldatensatz in getrennten Ressourcen halten.

Diese Aktion extrahiert PDF-Text lokal ohne KI-Anfrage. Sie führt keine OCR auf gescannten Seiten aus. Ihre einzelne primitive Ausgabe wird direkt in der Ressource gespeichert: Verwenden Sie `extrahiert`, nicht `extrahiert.text`. Aktionen mit mehreren Rückgabeeigenschaften speichern dagegen eine JSON-Zeichenfolge.

Der Uploadauslöser läuft nach jedem erfolgreichen Anhang-Upload. Die Aktion durchsucht die Anhänge des Datensatzes: Ein ausdrücklicher Dateinamenfilter hat Vorrang; sonst bevorzugt sie einen Lebenslauf-/CV-/Resume-/Profil-Dateinamen oder die einzige PDF. Mehrere nicht eindeutig zuordenbare PDFs, ein fehlender Treffer oder ein Extraktionsfehler können ein leeres Ergebnis liefern. Ergänzen Sie vor dem Überschreiben eines bestehenden Zielwerts eine Entscheidung, die nicht leeren extrahierten Text verlangt. Das bloße Anhängen einer Datei ergänzt die globale Suche nicht um ihren Inhalt; erst das Speichern des extrahierten Texts in einem durchsuchbaren Datensatzfeld tut dies.

Eingehende E-Mails verarbeiten

Ein Flow für eingehende E-Mail verwendet das aktive Microsoft-365-Konto unter **Setup → Postausgang**. Er kann kein separates Postfach wählen. Konfigurieren Sie im Auslöserbereich Quellobjekt, Filterbedingungen und Ordner für verarbeitete Nachrichten. Filter für Absenderadresse, Absendername, Empfänger, Betreff und Text werden mit UND verknüpft; Textvergleiche ignorieren Groß- und Kleinschreibung.



1. Der Flow verwendet das aktive Microsoft-365-Konto unter Setup → Postausgang.
2. Vor Aktivierung dieses Entwurfs Bedingungen ergänzen.
3. Erfolgreiche Verarbeitung verschiebt die Nachricht in diesen Ordner.

Der gezeigte Entwurf hat noch keinen Filter. Ergänzen Sie vor der Aktivierung die vorgesehene Betreff- oder Absenderbedingung; ein leerer Filter passt auf alle eingehenden Nachrichten. Der deutsche Editor verwendet standardmäßig **Verarbeitet** als Zielordner; im englischen Editor heißt der Standard **Processed**.

Verwenden Sie in Flow-Ausdrücken den Kontext `$Email`, beispielsweise `$Email.betreff` für den Betreff. Ergänzen Sie Datensätze erstellen oder eine andere ausdrückliche Schreibaktion zum Speichern des beabsichtigten Ergebnisses. Der lokale Test glich

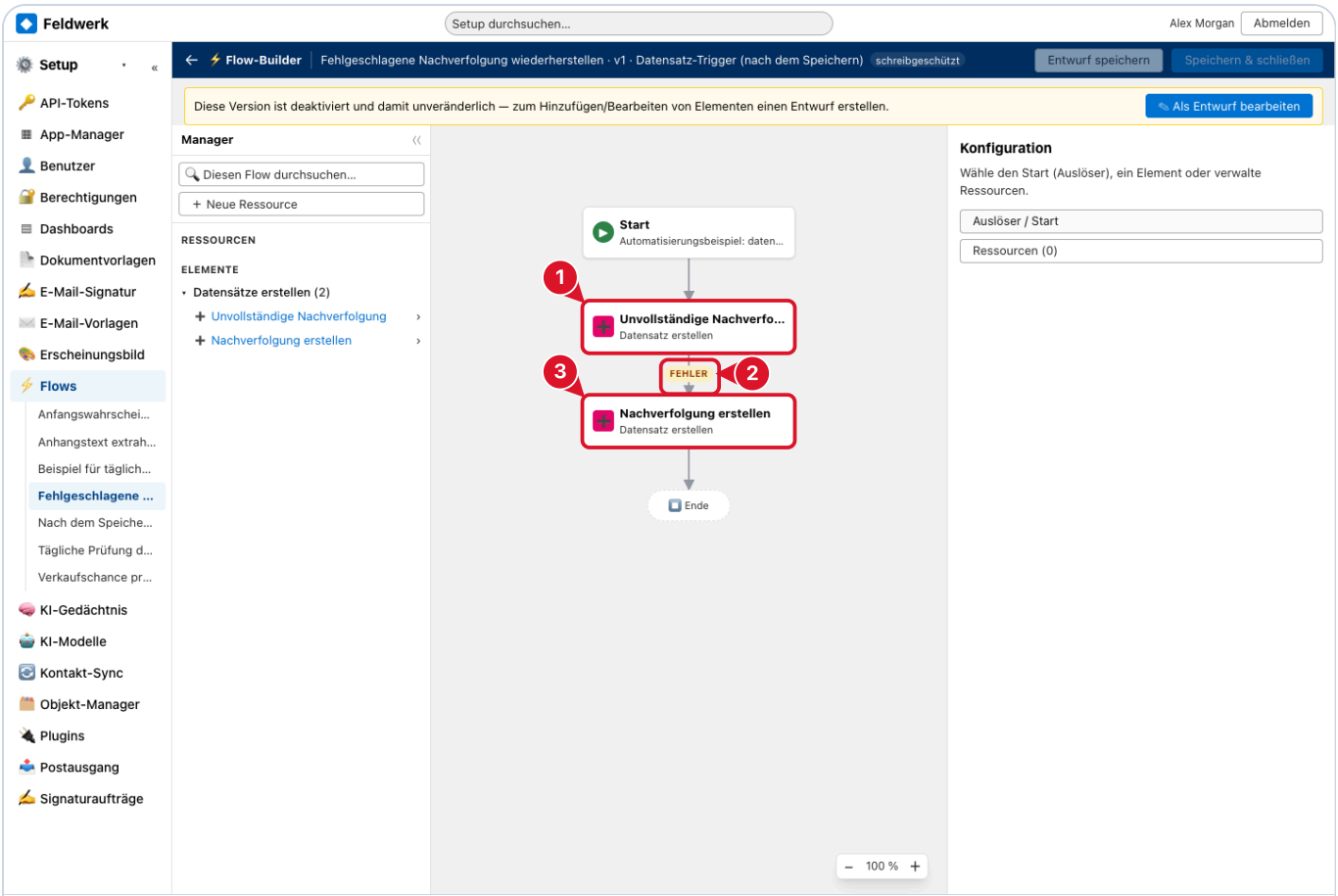
TRAINING gegen einen Betrefffilter training ab, erstellte einen Datensatz und ignorierte eine unpassende Nachricht. Wiederholtes Abfragen erzeugte kein Duplikat.

Nach erfolgreichem Abschluss aller passenden Flows wird die Nachricht aus dem Posteingang in den konfigurierten Verarbeitungsordner verschoben. Passen mehrere Flows, werden alle ausgeführt; der erste passende Flow bestimmt den Zielordner. Das ist von der Postfacherfassung getrennt, die keine Nachrichten verschiebt. Ein fehlgeschlagener Flow gibt die Verarbeitungssperre frei, sodass eine spätere Abfrage erneut versuchen kann. Frühere erfolgreiche Schreibvorgänge werden nicht automatisch zurückgerollt; gestalten Sie Wiederholungen daher ohne doppelte Wirkungen. Scheitert das Verschieben nach erfolgreicher Verarbeitung, bleibt die Sperre bestehen und die Flows werden nicht automatisch erneut ausgeführt.

Der Handbuchtest verwendet ein simuliertes Postfach und die echte Flow-Engine. Er belegt keine Zustellung oder Zugriffsberechtigung für ein reales Microsoft-Konto. Prüfen Sie Kontoverbindung und Ordnerverhalten, bevor Sie diese Automatisierung für echte E-Mails aktivieren.

Ein fehlgeschlagenes Element bewusst behandeln

Verbinden Sie den Fehlerzweig eines Elements mit einem Wiederherstellungsschritt, wenn der Ablauf einen sinnvollen Ersatzweg hat. Im geprüften Beispiel ließ Datensätze erstellen einen Pflichtnamen aus; sein Fehlerzweig erstellte stattdessen eine korrekt benannte Nachverfolgung. Der Ausführungsstatus lautete `ok_mit_fehlerbehandlung` (mit behandeltem Fehler abgeschlossen).



1. Der Schreibversuch scheitert an einem fehlenden Pflichtfeld.
2. Die Fehlerverbindung wählt den Ersatzweg.
3. Der Wiederherstellungsschritt erzeugt die gültige Nachverfolgung.

Ein Wiederherstellungspfad sollte genügend Informationen festhalten, um den ursprünglichen Fehler zu erkennen. Ein erfolgreich wirkender Abschlussbildschirm beweist nicht, dass jeder beabsichtigte Schreibvorgang erfolgreich war. Arbeiten nach dem Speichern erfolgen erst nach der ursprünglichen Datensatzspeicherung; ein späterer Fehler macht diese nicht rückgängig.

Ausführung testen und prüfen

Die Aktion **Debug** im Builder ist für Bildschirm-Flows verfügbar. Sie speichert zuerst den Entwurf und führt echte Aktionen aus; siehe Bildschirm-Flows ausführen. Automatische Flows testen Sie über ihr tatsächliches Ereignis: Datensatz speichern, Datei hochladen, geplantes Zeitfenster verarbeiten oder passende Nachricht empfangen.

Derzeit gibt es keine Oberfläche für den Ausführungsverlauf. Administratoren können die letzten 50 Ausführungsdatensätze einer Version über `GET /api/fluss-versionen/{versionId}/ausfuehrungen` prüfen. Kontrollieren Sie Status, Fehler,

Elementprotokoll und Dauer und anschließend den entstandenen Datensatz, Anhang oder Postausgangseintrag. Zum angemeldeten Zugriff siehe API und MCP.

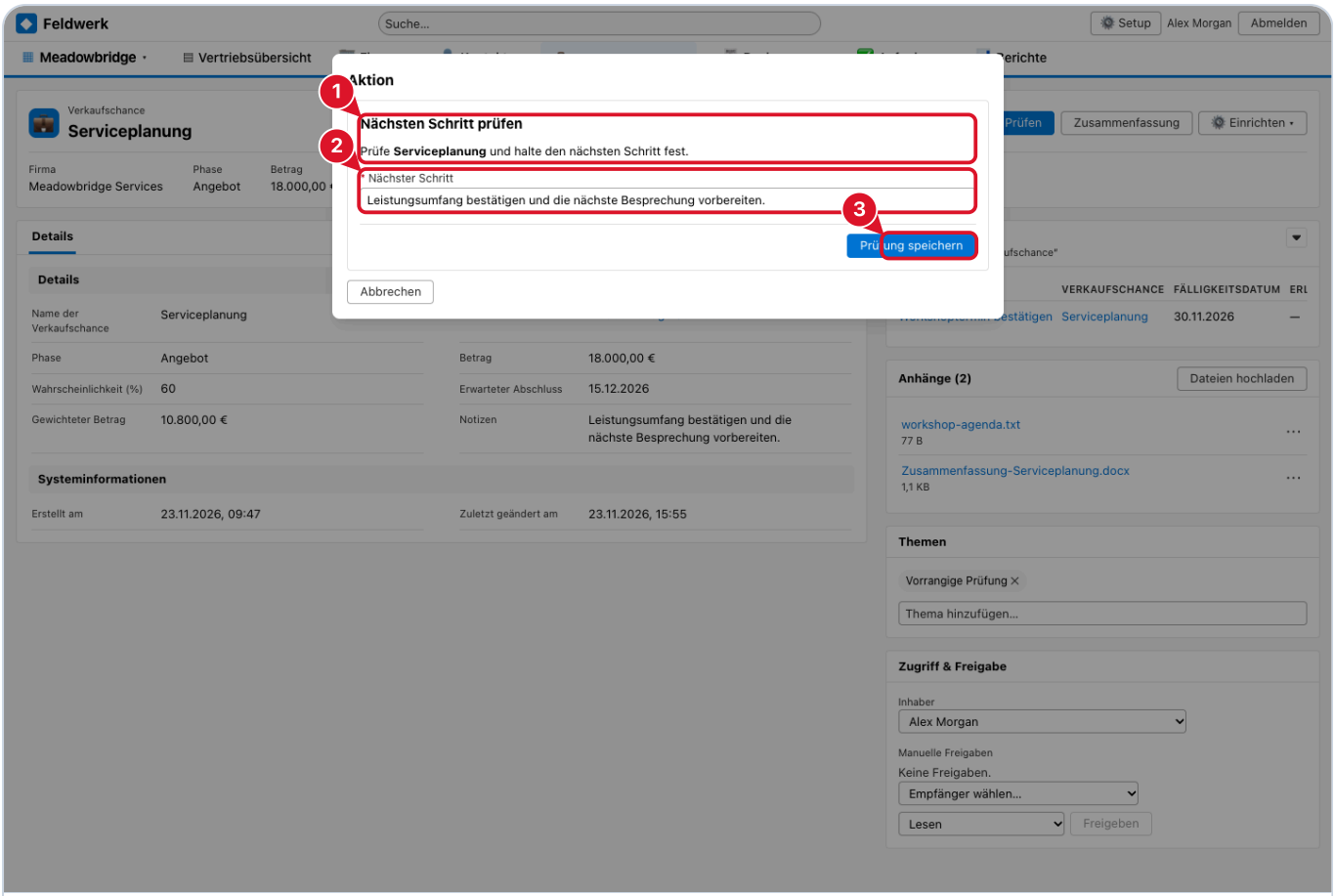
Bildschirm-Flows

Geführte Aufgaben ausführen und Bildschirme gestalten, die das gewünschte Ergebnis speichern.

Eine geführte Aufgabe starten

Ein Bildschirm-Flow erfragt Eingaben und führt zwischen Bildschirmen konfigurierte Logik aus. Er kann über eine Datensatzaktion gestartet, auf einer Datensatzseite eingebettet oder in einem Dashboard platziert werden. Der Startpunkt bestimmt, ob ein aktueller Datensatz verfügbar ist.

Im Handbuch startet die Aktion **Prüfen** an der Verkaufschance einen Flow mit Serviceplanung als Datensatzkontext. Der erste Bildschirm zeigt den Namen der Verkaufschance und die bearbeitbare Eingabe Nächster Schritt.



1. Datensatz und Anleitung prüfen.
2. Nächsten Schritt eingeben.
3. Zu den konfigurierten Speicher- und Abschlussschritten fortfahren.

1. Öffnen Sie die vorgesehene Verkaufschance und wählen Sie Prüfen.
2. Lesen Sie die Anleitung und prüfen Sie den aktuellen Datensatz.
3. Geben Sie den nächsten Schritt ein und wählen Sie **Prüfung speichern**.
4. Warten Sie auf den Bestätigungsbildschirm und beenden Sie den Flow.
5. Aktualisieren oder öffnen Sie die Verkaufschance erneut und prüfen Sie den gespeicherten Notizwert.

Navigation und Speicherung

Weiter, Zurück und Abschlussaktionen hängen vom Flow ab. Pflichteingaben und konfigurierte Validierungen müssen vor dem Fortfahren erfüllt sein. Zurück ist keine allgemeine Rückgängig-Schaltfläche: Nach einem Schreibvorgang oder anderen

unumkehrbaren Effekten kann die Laufzeit die Rückkehr zu einem früheren Bildschirm verhindern.

Abbrechen schließt oder verlässt die Interaktion, verspricht aber nicht, bereits ausgeführte Arbeit rückgängig zu machen. Erscheint nach einer Aktion ein Fehler, prüfen Sie vor einer Wiederholung den tatsächlichen Datensatz- oder Postausgangsstand.

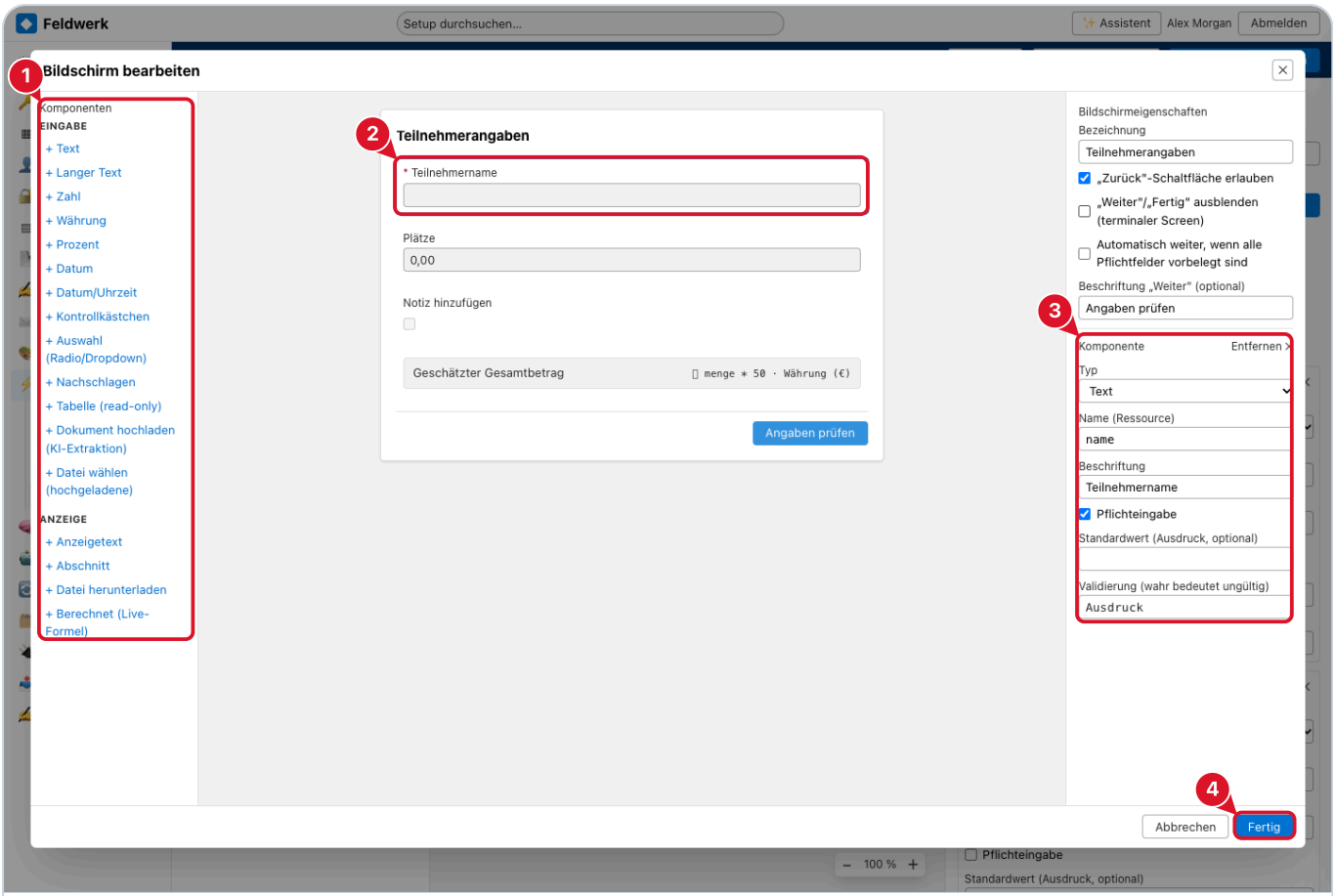
Das Prüfbeispiel aufbauen

1. Erstellen Sie einen Bildschirm-Flow für Verkaufschancen.
2. Ergänzen Sie eine Textressource für den nächsten Schritt und eine Datensatzressource für die Ziel-Verkaufschance.
3. Ergänzen Sie einen Bildschirm mit Anleitung und Pflichttexteingabe. Verwenden Sie die Notizen des aktuellen Datensatzes als Anfangswert.
4. Fügen Sie Datensätze abrufen mit Filter auf die aktuelle Datensatz-ID hinzu und speichern Sie das Ergebnis in der Zielressource.
5. Aktualisieren Sie mit Datensätze aktualisieren diese Ressource und schreiben Sie die Texteingabe in Notizen.
6. Ergänzen Sie einen Abschlussbildschirm und verbinden Sie die Elemente der Reihe nach.
7. Speichern, testen und aktivieren Sie den Flow und ordnen Sie ihn der Datensatzaktion zu.

Der Handbuchtest ändert Notizen, liest den neuen Wert über die API und stellt anschließend den Anfangswert wieder her. Damit wird das gespeicherte Ergebnis geprüft, nicht nur der Bildschirmtext.

Einen Bildschirm visuell bearbeiten

Wählen Sie ein Bildschirmelement im Ablauf und anschließend **Bildschirm bearbeiten (WYSIWYG)**. Fügen Sie links Komponenten hinzu, wählen Sie eine in der Vorschau und konfigurieren Sie rechts ihre Eigenschaften. Name bezeichnet die Ressource für spätere Ausdrücke; Beschriftung ist der sichtbare Benutzertext. Kehren Sie mit Fertig zum Ablauf zurück und speichern Sie anschließend den Flow-Entwurf.



1. Benötigte Eingabe- oder Anzeigekomponente hinzufügen.
2. Komponente in der Vorschau wählen.
3. Ressourcenname, Beschriftung, Pflichtstatus und Validierung konfigurieren.
4. Bildschirmänderungen übernehmen und anschließend den Flow-Entwurf speichern.

Der visuelle Editor ist eine Layoutvorschau. Seine Beispielwerte beweisen nicht, dass ein Standardausdruck oder eine Berechnung zur Laufzeit funktioniert. Testen Sie die tatsächliche Version über **Debug** im Flow-Builder. Debug speichert zuerst den Entwurf und führt echte Schreibvorgänge, Downloads und konfigurierte Aktionen aus.

Bildschirmkomponenten wählen

Verwenden Sie unterstützte Eingabekomponenten für Text, Zahlen, Datumswerte, Auswahlen und Nachschlagefelder. Anzeigetext kann die Aufgabe erklären und unterstützte Platzhalterwerte enthalten. Tabellen und berechnete Anzeigen helfen bei der Prüfung von Informationen. Testen Sie Sichtbarkeitsbedingungen und Komponentenvalidierungen mit passenden und nicht passenden Eingaben.

Benennen Sie Ressourcen einheitlich und wählen Sie kompatible Datentypen. Der Wert einer Bildschirmkomponente muss von der späteren Zuweisung oder dem schreibenden Datensatzelement verwendet werden. Eine Eingabe auf einem Bildschirm speichert ihren Wert allein noch nicht in einem Datensatz.

Komponente	Verwendung und Konfiguration
Anzeigetext / Abschnitt	Aufgabe erklären oder Bildschirm in Abschnitte teilen. Text unterstützt konfigurierte Platzhalterwerte.
Text / Langtext	Kurzen Wert oder mehrzeilige Notiz erfassen. Bei Bedarf Pflichtregel und klare Validierungsmeldung ergänzen.
Zahl / Währung / Prozent	Numerische Werte erfassen. Typ passend zu Ressource und Zielfeld wählen; ein sichtbares Währungs- oder Prozentsuffix speichert für sich kein Feld.
Datum / Datum-Uhrzeit	Datum oder Datum mit Uhrzeit erfassen. Gespeicherten Wert einschließlich der Datums-/Uhrzeitinterpretation des Arbeitsbereichs prüfen.
Checkbox	Ja/Nein-Wert erfassen und später in einer Entscheidung oder der Sichtbarkeitsregel eines Folgebildschirms verwenden.
Auswahl	Gespeicherte Werte und Anzeigebeschriftungen direkt konfigurieren oder unterstützte feldbasierte Optionen verwenden. Direkt konfigurierte Optionen haben Vorrang. Die Benutzerbeschriftung kann vom gespeicherten Wert abweichen.
Nachschlagefeld	Zielobjekt wählen und Benutzer dessen zugängliche Datensätze durchsuchen lassen. Eine Nachschlageauswahl erteilt keine Datensatzberechtigung.

Komponente	Verwendung und Konfiguration
Tabelle	Mit Datensätze abrufen Datensätze in eine Sammlung laden, diese als Tabellenquelle wählen und Spalten konfigurieren. Einzelauswahl liefert einen gewählten Datensatz; Mehrfachauswahl eine Sammlung. Optionale Vorauswahl aller Zeilen gilt bei Mehrfachauswahl.
Berechnet	Ein auf aktuelle Eingaben reagierendes Formelergebnis anzeigen. Text-, Zahlen-, Währungs- oder Prozentformatierung wählen. Zum Speichern ein späteres Schreibelement verwenden.
Dokument-Upload / Dateiauswahl	Dokumente für konfigurierte KI-Extraktion hochladen und optional daraus Dateien wählen. Die Dateiauswahl bezieht sich auf während dieses Interviews hochgeladene Dateien; sie durchsucht nicht alle Anhänge des Arbeitsbereichs.
Datei herunterladen	Browserdownload aus Inhalts- und Dateinamenausdrücken erzeugen. Erstellt nicht automatisch einen Anhang.

Bei einer Auswahltabelle akzeptiert der Server nur Zeilen aus ihrer konfigurierten Quellsammlung. Eine gewählte Zeile wird nicht allein durch ihre Auswahl gespeichert. Verwenden Sie den ausgewählten Datensatz oder die Sammlung ausdrücklich in der späteren Flow-Logik.

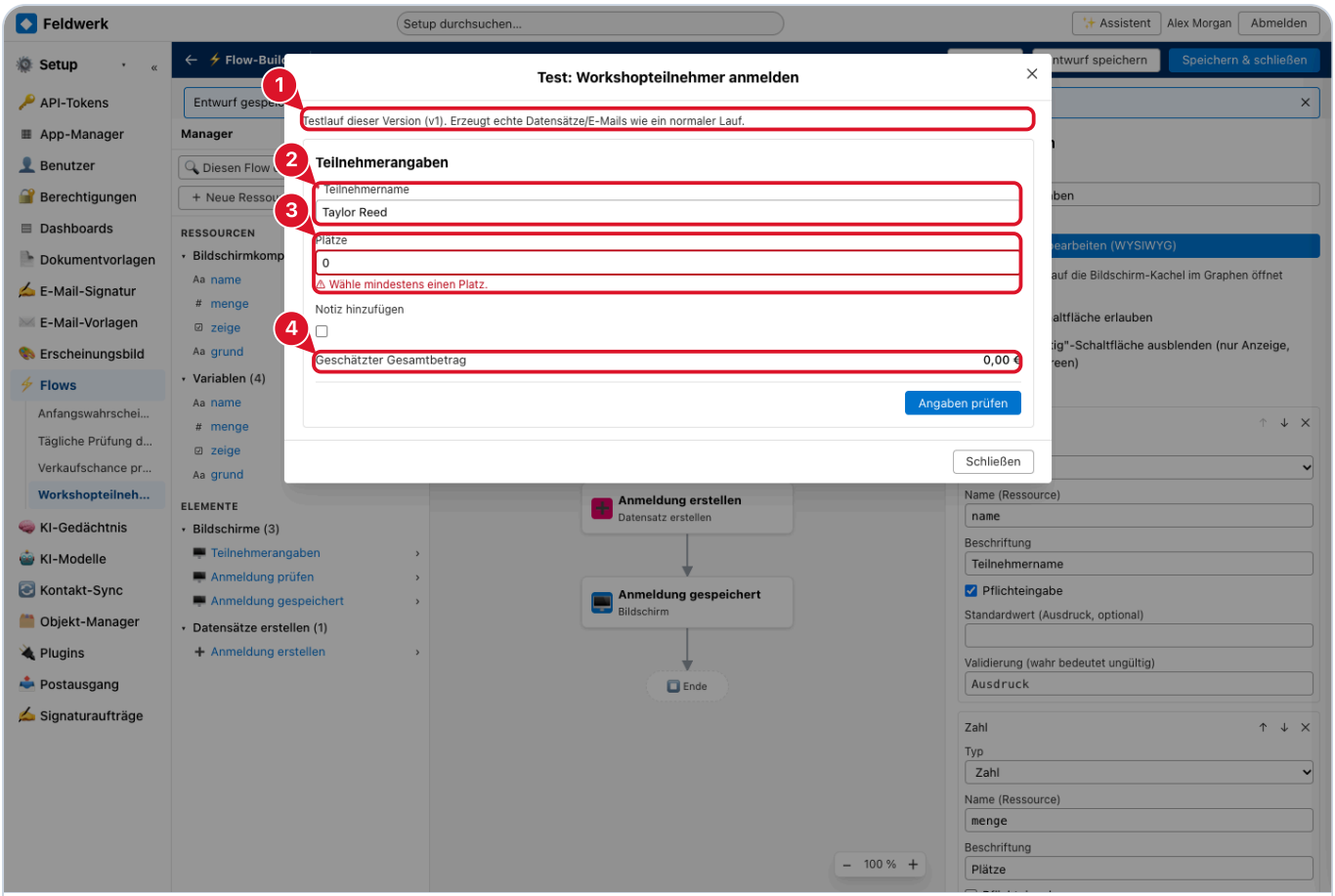
Dokument-Uploads und Speicherung als Anhang

Dokument-Upload kann mit dem konfigurierten KI-Anbieter und Zielobjektschema Werte in Flow-Ressourcen extrahieren. Prüfen Sie die Vorschläge vor dem Fortfahren; Extraktion ersetzt kein schreibendes Datensatzelement. Dateien bleiben im laufenden Browser, bis der Flow Datensätze erzeugt und die Anhang-Uploadphase erreicht.

Eine Anhangzuordnung kann eine Datei zu einem bestimmten neu erzeugten Datensatz leiten. Ohne konkrete Zuordnung gehen übrige Dateien an das konfigurierte Ersatzobjekt oder den zuerst erzeugten Datensatz. Wird kein Datensatz erzeugt, begründet die Dateiauswahl allein keinen gespeicherten Anhang. Prüfen Sie den resultierenden Dateibereich und Uploadwarnungen. Ein Datensatz kann erfolgreich gespeichert worden sein, obwohl sein späterer Anhang-Upload scheiterte.

Beispiel: eine Workshopanmeldung validieren

1. Ergänzen Sie eine Pflichttextkomponente namens `name` mit der Beschriftung Teilnehmername.
2. Ergänzen Sie eine Zahlenkomponente namens `menge` mit der Beschriftung Plätze und dem Standardausdruck `1`.
3. Setzen Sie ihre Validierungsformel auf `menge < 1` und die Meldung auf **Wählen Sie mindestens einen Platz.** Ein wahres Validierungsergebnis bedeutet eine ungültige Eingabe.
4. Ergänzen Sie eine berechnete Komponente mit Ausdruck `menge * 50` und Währungsformat. Im geprüften Beispiel zeigt die Eingabe 3 sofort 150,00 €.
5. Starten Sie Debug, lassen Sie den Namen leer und fahren Sie fort. Die Pflichtfeldmeldung erscheint, ohne das Formular zu schließen. Geben Sie einen Namen und 0 Plätze ein, prüfen Sie die eigene Fehlermeldung und korrigieren Sie den Wert.



1. Debug führt echte Aktionen dieser Version aus.
2. Der eingegebene Teilnehmername bleibt erhalten.
3. Vor dem Fortfahren die Platzanzahl korrigieren.
4. Die berechnete Summe reagiert auf die Eingabe.

Automatische Pflichtfeldmeldungen folgen der Benutzersprache. Eigene Validierungsmeldungen sind Inhalte des Flow-Autors und bleiben wie konfiguriert. Eine berechnete Anzeige zeigt eine Ergebnisvorschau; sie aktualisiert für sich weder Feld noch Ressource.

Bedingte Bildschirme und Zurück

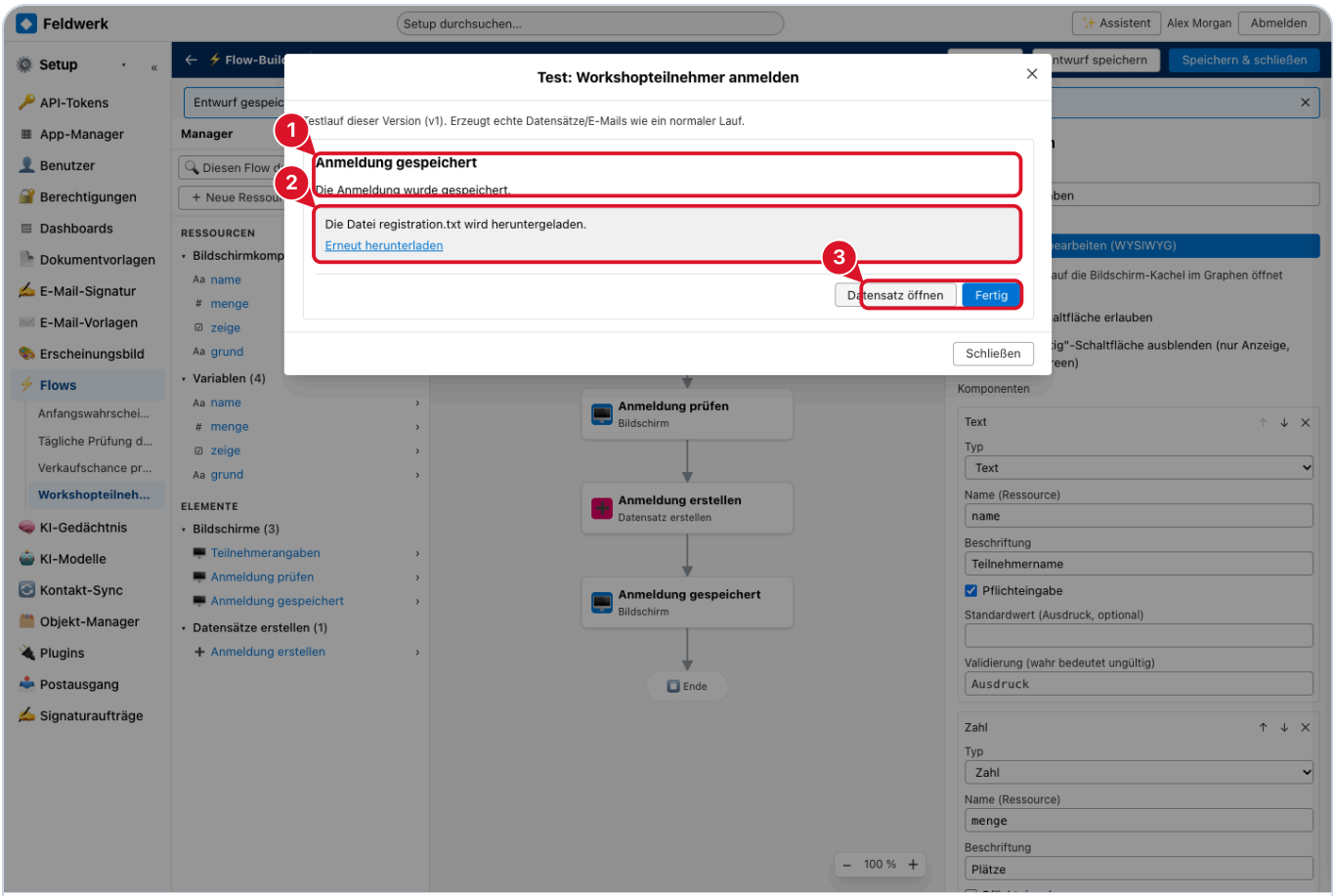
Der geprüfte Flow besitzt auf dem ersten Bildschirm eine Checkbox Notiz hinzufügen. Auf dem Prüfbildschirm ist Anmeldenotiz nur dann sichtbar und Pflicht, wenn diese Checkbox wahr ist. Fahren Sie ohne Auswahl fort, um zu prüfen, dass die versteckte Pflichtkomponente den Flow nicht blockiert. Kehren Sie mit Zurück zurück, wählen Sie die Checkbox und fahren Sie erneut fort, um den sichtbaren Fall zu prüfen.

Sichtbarkeits- und bedingte Pflichtregeln werden beim serverseitigen Rendern eines Bildschirms ausgewertet, auch nach dem Absenden. Gehen Sie nicht davon aus, dass eine Checkbox die Sichtbarkeit auf demselben Bildschirm sofort ändert. Platzieren Sie abhängige Fragen auf dem nächsten Bildschirm, wenn ein verlässlicher Übergang nötig ist. Berechnete Anzeigen aktualisieren sich dagegen bereits beim Tippen.

Zurück behält bereits von einem früheren Bildschirm übermittelte Werte. Änderungen, die auf dem verlassenen Bildschirm gerade bearbeitet werden, werden dabei nicht übermittelt. Zurück wird nur angeboten, wenn es aktiviert ist, ein vorheriger Bildschirm existiert und kein Schreibvorgang oder anderer geschützter Effekt stattgefunden hat. Das Schließen der Interaktion macht einen gespeicherten Datensatz nicht rückgängig.

Einen Flow abschließen und eine erzeugte Datei herunterladen

Das Workshopbeispiel verwendet nach dem Prüfbildschirm Datensätze erstellen. Sein Abschlussbildschirm bestätigt die Speicherung und enthält eine Komponente Datei herunterladen. Konfigurieren Sie Inhalt als Ausdruck wie `"Registration: " & name`, Dateinamen als `"registration.txt"` und MIME-Typ als `text/plain`.



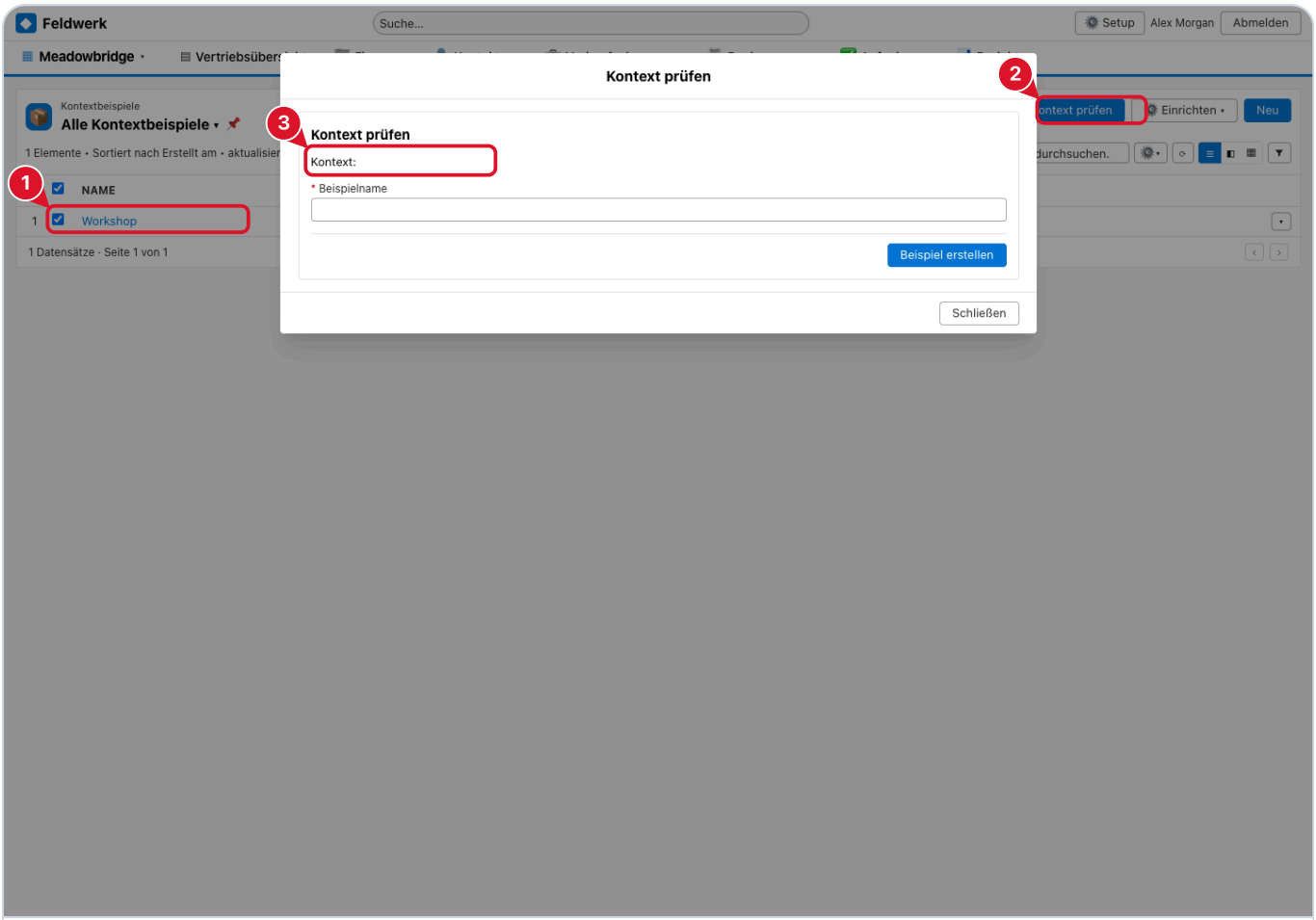
1. Konfigurierte Abschlussmeldung lesen.
2. Erzeugte Datei abrufen oder erneut herunterladen.
3. Tatsächlichen Datensatz zur Prüfung öffnen oder die Interaktion beenden.

Der Browser lädt den erzeugten Text beim Anzeigen der Komponente herunter; Erneut herunterladen wiederholt den Download. Das erzeugt einen Browserdownload, keinen gespeicherten Datensatzanhang. Im geprüften Beispiel enthielt registration.txt den Text Registration: Taylor Reed; ein separater API-Abruf bestätigte den neuen Datensatz. Prüfen Sie ihn mit Datensatz öffnen oder schließen Sie mit Fertig ab. Eingebettete Widgets können nach Abschluss ein neues Interview starten.

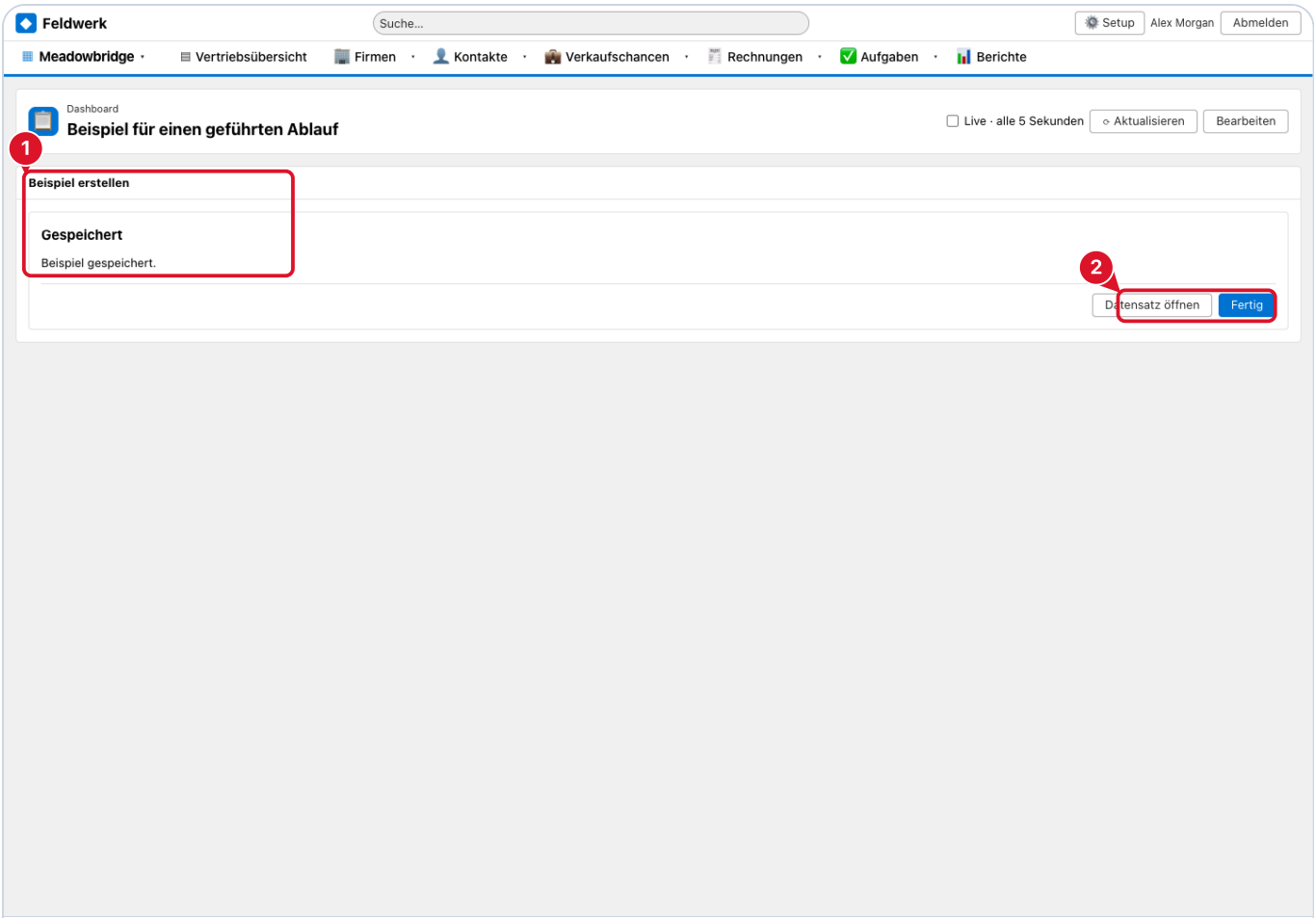
Datensatzkontext und Benutzerzugriff

Startpunkt	Datensatzkontext
Aktion oder Flow-Widget auf einer Datensatzseite	Der aktuelle Datensatz wird übergeben.
Flow im Aktionsmenü einer Listenzeile	Der Datensatz dieser Zeile wird übergeben.

Startpunkt	Datensatzkontext
Flow-Schaltfläche im Listenkopf	Kein Datensatz wird übergeben, auch wenn Zeilen angekreuzt sind.
Flow-Widget im Dashboard	Kein Datensatz wird übergeben. Der Flow muss sein Ziel erfragen oder finden.



1. Eine angekreuzte Zeile wird nicht zum Flow-Kontext.
2. Die Schaltfläche im Listenkopf startet das Interview.
3. Der Kontext bleibt leer; erfragen Sie bei Bedarf ein Ziel.



1. Abschlussmeldung des eingebetteten Flows prüfen.
2. Erzeugten Datensatz öffnen oder mit Fertig ein neues Interview starten.

Ein Flow, der einen aktuellen Datensatz erwartet, muss Startpunkte ohne Datensatz behandeln. Im Anzeigetext zeigt `Kontext: {!$Datensatz.name}` den aktuellen Datensatznamen, sofern einer übergeben wird; das Dollarzeichen ist erforderlich. Fragen Sie beispielsweise vor Datensätze abrufen nach einer Nachschlageauswahl, statt ein gefülltes `$Datensatz` vorauszusetzen. Ein Kontextdatensatz aus einem anderen Objekt wird abgelehnt. Dashboardwidgets können nach Fertig ein neues Interview starten; verwechseln Sie zuvor übermittelte Werte nicht mit dessen neuen Eingaben.

Bildschirm-Flows laufen mit den Berechtigungen der ausführenden Person. Der Start mit einem Kontextdatensatz benötigt Leserecht darauf. Leserecht allein erlaubt keinen Schritt Datensätze erstellen oder Datensätze aktualisieren. Das Testen einer bestimmten Entwurfsversion über Debug ist Administratoren vorbehalten. Ein Interview gehört dem Benutzer, der es gestartet hat; andere Benutzer können es nicht allein mit Kenntnis seiner Lauf-ID übernehmen. Ein für Administratoren funktionierender Flow kann bei einem Mitglied scheitern, das ein Nachschlageziel nicht lesen oder ein Feld nicht bearbeiten darf. Testen Sie mit dem vorgesehenen Konto und prüfen Sie jeden Lese- und Schreibzugriff des Flows.

Öffentliche Erfassung ist eine separate Funktion

Verwenden Sie für anonyme Erfassung mit einem einzelnen Formular ein öffentliches Webformular. Ein angemeldeter Bildschirm-Flow ist nicht allein wegen seiner URL ein öffentlicher Endpunkt. Entwürfe öffentlicher Bildschirm-Flows in Planungsdokumenten belegen keine verfügbare Funktion.

Einen stockenden Flow untersuchen

Prüfen Sie aktive Version, richtige Definition in der Startaktion und gültigen aktuellen Datensatz. Prüfen Sie bei einem fehlgeschlagenen Schritt die Ausführungsinformationen der Version und den Benutzerzugriff. Hat ein abgeschlossener Flow keinen Datensatz geändert, prüfen Sie zuerst Zielressource und Feldzuordnung.

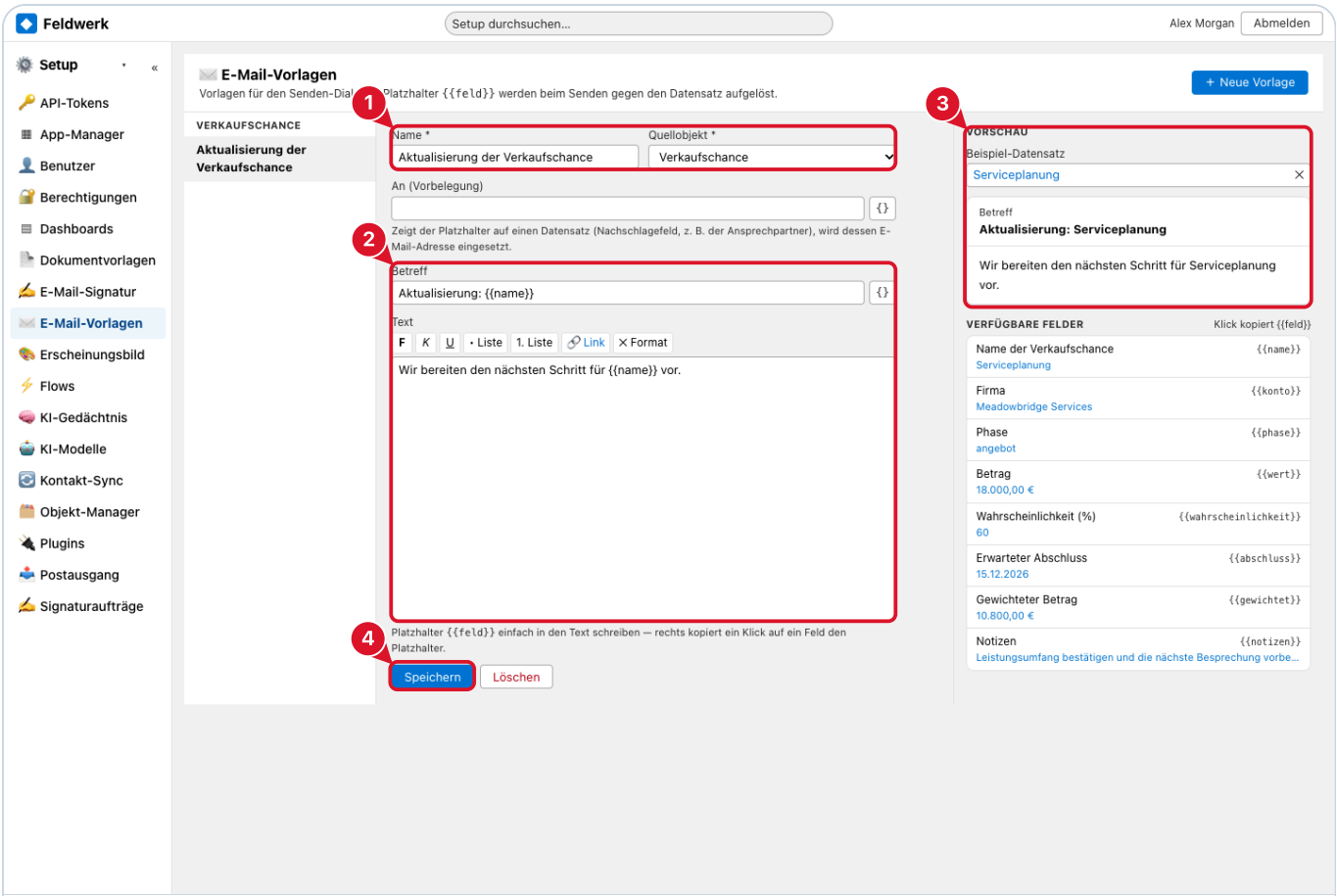
E-Mail-Vorlagen, Signaturen und Versand

Datensatzbezogene Nachrichten vorbereiten, Empfänger prüfen und den Versandstatus verfolgen.

Vor dem Senden

Die E-Mail-Verfügbarkeit hängt von installierten Aktionen, Ihren Berechtigungen und der konfigurierten ausgehenden Verbindung der Instanz ab. Eine Vorlage bereitet Inhalte vor; sie richtet keine Mailverbindung ein. Administratoren verwalten die Versandeinstellungen und prüfen **Setup → Postausgang**.

Verwenden Sie zum Lernen fiktive Empfänger oder ein von der Administration kontrolliertes Testpostfach. Eine erfolgreich eingereihte Nachricht beweist noch nicht, dass sie beim Empfänger angekommen ist. Prüfen Sie den Versandstatus, bevor Sie eine Nachricht erneut senden.



1. Vorlage benennen und Quellobjekt wählen.
2. Betreff und Inhalt mit Feldplatzhaltern schreiben.
3. Aufgelöste Beispielvorschau prüfen.
4. Vorlage speichern.

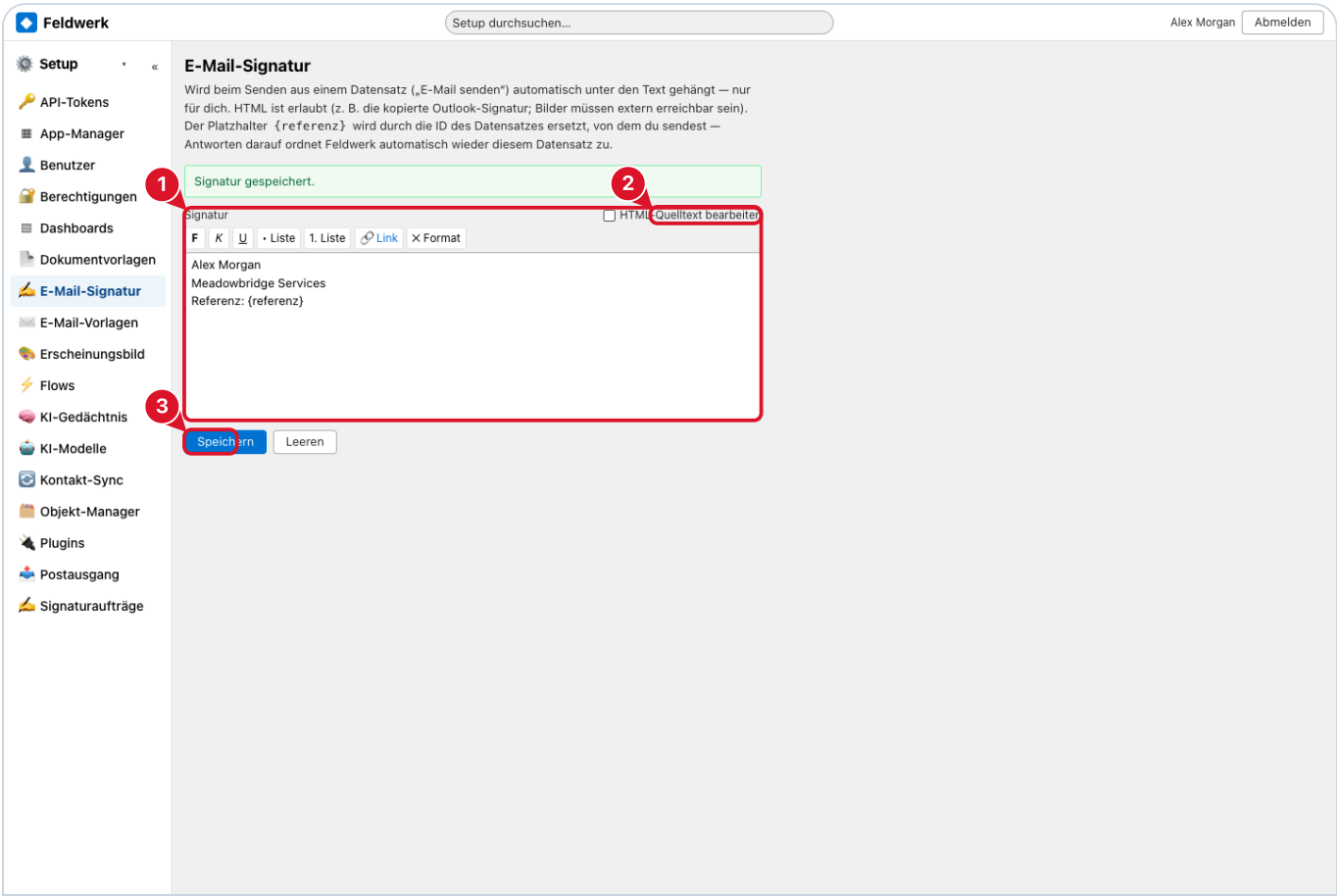
Eine wiederverwendbare Vorlage erstellen

1. Öffnen Sie **Setup → E-Mail-Vorlagen** und wählen Sie **Neue Vorlage**.
2. Geben Sie einen aussagekräftigen Namen ein und wählen Sie das Quellobjekt, etwa Verkaufschancen.
3. Schreiben Sie Betreff und Inhalt. Kopieren Sie genaue Platzhalter aus der Liste verfügbarer Felder, statt Feldbeschriftungen einzutippen.
4. Konfigurieren Sie das Empfängerfeld oder den Empfängerausdruck, wenn die Vorlage eine Adresse vorbelegen soll.
5. Wählen Sie einen Beispieldatensatz und prüfen Sie die **Vorschau**.
6. Speichern Sie die Vorlage und prüfen Sie sie anschließend von einem Datensatz des

Quellobjekts aus.

Platzhalter verwenden Feld-API-Namen wie `{{name}}`. Bei der Verkaufschance Serviceplanung wird „Aktualisierung: {{name}}“ zu „Aktualisierung: Serviceplanung“. Die Vorschau löst Inhalte anhand des ausgewählten Beispieldatensatzes auf. Ohne Beispiel bleiben Platzhalter in der Vorschau leer; das beweist nicht, dass die Vorlage einsatzbereit ist.

Prüfen Sie erforderliche Werte und den aufgelösten Empfänger getrennt. Ein Kontaktname ohne E-Mail-Adresse ist kein gültiges Versandziel. Eine Vorlagenänderung verändert bereits eingereihte Nachrichten nicht nachträglich.



1. Signatur bearbeiten.
2. Bei Bedarf zum HTML-Quelltext wechseln.
3. Für Ihr Konto speichern.

Ihre persönliche Signatur festlegen

1. Öffnen Sie **Setup → E-Mail-Signatur**.

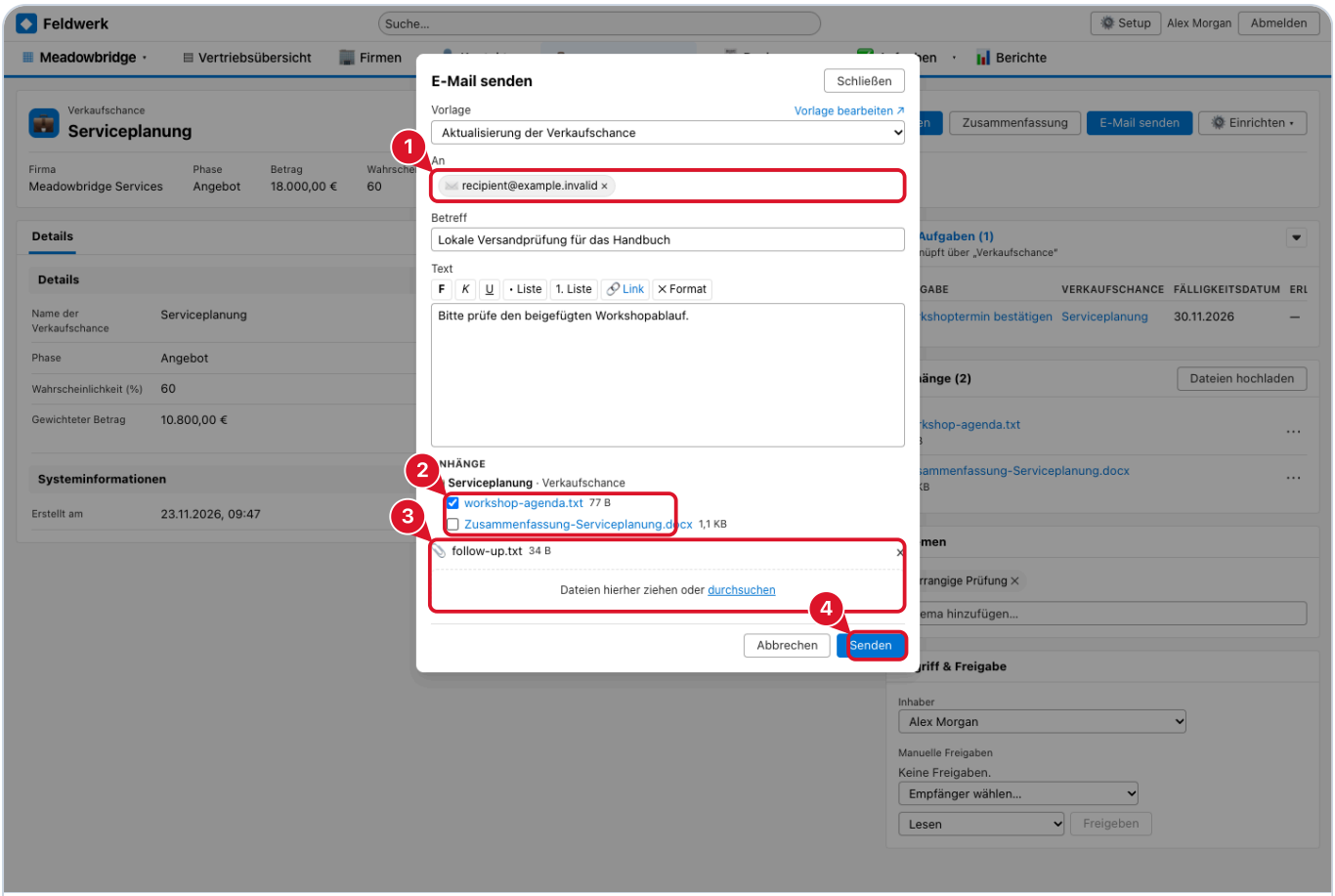
2. Geben Sie Ihre Signatur im visuellen Editor ein oder wählen Sie **HTML-Quelltext bearbeiten**, um ihr HTML zu pflegen.
3. Verwenden Sie extern gehostete Bilder, falls die Signatur Bilder enthält.
4. Wählen Sie **Speichern**.

Die Signatur gehört zu Ihrem Konto und wird beim Senden aus einem Datensatz angehängt. Der exakte Platzhalter `{referenz}` wird durch die öffentliche ID des auslösenden Datensatzes ersetzt. Übersetzen Sie diesen technischen Platzhalter nicht. Er hilft einer konfigurierten E-Mail-Erfassung, Antworten einem Datensatz zuzuordnen.

Aus einem Datensatz senden

1. Öffnen Sie den vorgesehenen Datensatz und wählen Sie seine Aktion **E-Mail senden**.
2. Wählen Sie eine verfügbare Vorlage oder verfassen Sie die Nachricht ohne Vorlage.
3. Prüfen Sie alle Empfänger, den aufgelösten Betreff und den Inhalt. Entfernen Sie falsche Empfängermarkierungen und ergänzen Sie den richtigen Kontakt oder die richtige Adresse.
4. Wählen Sie die für diese Nachricht benötigten Anhänge. Verfügbare Datensatzdateien und Dateien verwandter Listen hängen vom Zugriff ab; prüfen Sie tatsächliche Dateinamen und Inhalte.
5. Prüfen Sie Nachricht und Anhänge und wählen Sie **Senden**. Ihre gespeicherte persönliche Signatur wird vom Server angehängt; dieser Verfassungsdialog zeigt sie nicht in der Vorschau.
6. Warten Sie, bis sich der Dialog nach erfolgreichem Einreihen schließt, und prüfen Sie den Status der ausgehenden Nachricht. Ein eingebettetes E-Mail-Formular bleibt dagegen offen und bestätigt das Einreihen.

Geben Sie im Feld An einen Kontaktnamen ein und wählen Sie den passenden Vorschlag. Alternativ geben Sie eine vollständige Adresse ein und drücken Eingabe, um sie als Empfängermarkierung hinzuzufügen. Über deren Entfernen-Symbol korrigieren Sie die Empfängerliste. Beim Öffnen können vorhandene Dateien bereits ausgewählt sein. Prüfen Sie jedes Kästchen und wählen Sie alle Dateien ab, die nicht versendet werden sollen. Die Dateiauswahl fügt zusätzliche Dateien nur dieser Nachricht hinzu; sie werden nicht als neue Anhänge am Quelldatensatz gespeichert.



1. Tatsächliche Empfängeradresse prüfen und falsche Markierungen entfernen.
2. Nur für diese Nachricht vorgesehene Datensatzdateien ausgewählt lassen.
3. Zusatzdateien für diese Nachricht hinzufügen, ohne Datensatzanhänge anzulegen.
4. Senden reiht die Nachricht zur Hintergrundverarbeitung ein.

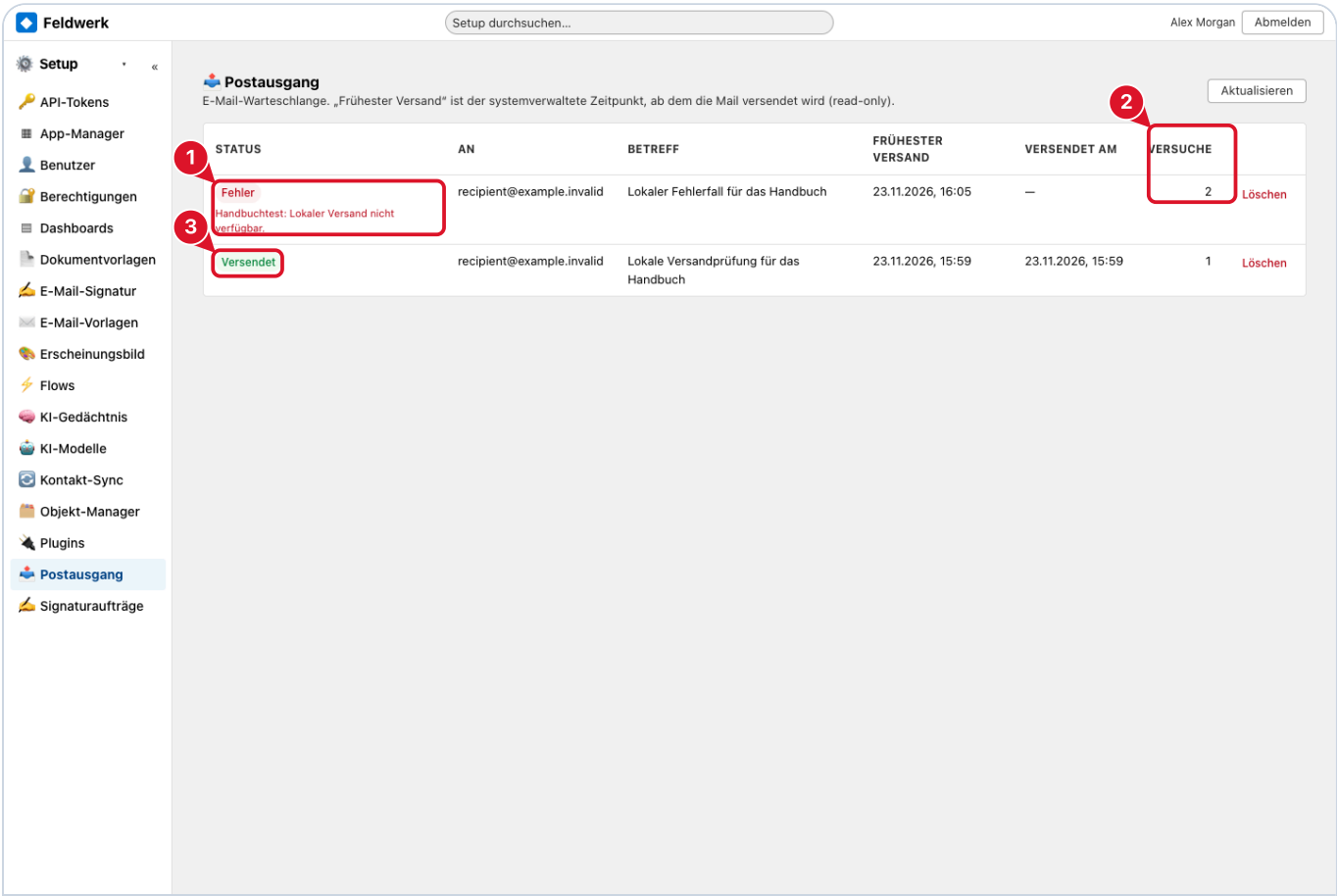
Hat der Datensatz keine geeignete Vorlage, kann die Administration im Setup eine erstellen. Fehlt ein Anhang, prüfen Sie seine Zugehörigkeit zu einem zugänglichen Datensatz und seine Verfügbarkeit für die Versandaktion. Laden Sie keine zweite Kopie hoch, nur weil die Auswahlliste noch nicht aktualisiert wurde.

Den Versand über die Warteschlange verstehen

Feldwerk reiht Nachrichten für den Hintergrundversand ein. Erfolgreiches Einreihen bedeutet, dass die Nachricht zur Verarbeitung angenommen wurde. Ausgangsdatensätze unterscheiden zwischen eingereiht, gesendet und fehlgeschlagen. Administratoren können Fehlerdetails und die konfigurierte Mailverbindung prüfen. Eine Entwicklungsinstanz kann einen Protokolltransport verwenden, der keine externen E-Mails zustellt.

Prüfen Sie vor einem erneuten Senden nach einem Fehler, ob die ursprüngliche Nachricht eingereiht oder gesendet wurde. Eine wiederholte Benutzeraktion kann eine weitere Nachricht erzeugen. Prüfen Sie bei automatisierten Nachrichten neben dem Postausgang auch die zugehörige Flow-Ausführung. Siehe Flows und Automatisierung.

Prüfen Sie unter **Setup → Postausgang** die Angaben **Status**, **Versuche**, **Frühester Versand** und **Gesendet am**. Nach einem vorübergehenden Transportfehler kann eine offene Nachricht einen zukünftigen frühesten Versandzeitpunkt haben, während sie auf den nächsten Versuch wartet. Nach Erreichen der konfigurierten Versuchsgrenze wird sie als Fehlgeschlagen markiert und zeigt den letzten Fehler. Frühester Versand wird vom System verwaltet und ist keine bearbeitbare Zeitplansteuerung.



1. Vor einem neuen Versuch den Fehlergrund lesen.
2. Versuche zählt die Versandversuche des Workers; dieser Test endete nach zwei Fehlschlägen.
3. Versendet bedeutet, dass der konfigurierte Transport abgeschlossen wurde. Dieser Screenshot verwendet einen lokalen Protokolltransport und belegt keine Zustellung an ein externes Postfach.

Eingehende Nachrichten und Antworten

Die Erfassung benötigt ein verbundenes Microsoft-365-Postfach und die passende Erfassungsoption am Datensatz der ausgehenden Verbindung. Eine Verbindung allein aktiviert die Erfassung nicht. Administratoren aktivieren **Eingehende E-Mails erfassen** für den Posteingang und separat **Gesendete E-Mails erfassen** für Gesendete Elemente. Die Erfassung speichert passende Nachrichten als Datensätze; sie verschiebt diese selbst nicht aus dem Posteingang. E-Mail-ausgelöste Flows werden separat konfiguriert und können verarbeitete Nachrichten verschieben.

Bei einer eingehenden Nachricht gleicht Feldwerk die Absenderadresse mit dem E-Mail-Feld der Kontakte ab, ersatzweise mit Leads. Bei Gesendeten Elementen wird stattdessen ein Empfänger abgeglichen. Unabhängig davon kann die erste vorhandene öffentliche Datensatz-ID in Betreff oder Inhalt den zugehörigen Geschäftsdatensatz liefern. Im Handbuchbeispiel identifiziert Jordan Reed den Kontakt und die Serviceplanung-Referenz die Verkaufschance. Eine gültige Referenz kann auch ohne passenden Kontakt eine Beziehung herstellen.

Eine Nachricht ohne passende Person und ohne gültige Datensatzreferenz wird nicht automatisch erfasst. Die Erfassung prüft Nachrichten-IDs, um wiederholte Importe derselben Nachricht zu verhindern. Über Feldwerk gesendete Nachrichten haben bereits Ausgangsprotokolle und werden von einer doppelten Erfassung aus Gesendeten Elementen ausgeschlossen. Prüfen Sie nach der Aktivierung die tatsächlichen Kontakt- und Geschäftsdatensatzverknüpfungen; eine zitierte Referenz kann auf einen anderen als den beabsichtigten Datensatz verweisen.

Öffnungsverfolgung verstehen

Wenn die Funktion aktiv und eine öffentliche Anwendungs-URL konfiguriert ist, erhalten ausgehende HTML-Nachrichten ein kleines Trackingbild. Die Verbindungsoption **Öffnungsverfolgung deaktivieren** schaltet dies ab. Ein Bildabruf nach erfolgreichem Versand aktualisiert **Geöffnet am** und **Öffnungen** am Ausgangs-E-Mail-Datensatz. Der erste Öffnungszeitpunkt bleibt erhalten; weitere Abrufe erhöhen die Anzahl.

Gemessen werden Bildabrufe, nicht bestätigtes Lesen durch Menschen. Blockierte Bilder können ohne Ereignis bleiben; automatische Bildabrufe können ein Ereignis ohne bewusstes Lesen erzeugen. Verstehen Sie die Öffnungsanzahl weder als Zustellbestätigung noch als Zahl unterschiedlicher Leser. Im lokalen Handbuchtest erzeugten zwei Bildabrufe zwei Öffnungen; kein externes Postfach wurde kontaktiert.

Häufige Probleme

Leerer eingesetzter Text bedeutet meist einen fehlenden Quellwert oder einen falschen Beispieldatensatz. Eine nicht verfügbare Vorlage kann zu einem anderen Quellobjekt gehören. Bleibt eine Nachricht eingereiht, müssen Versandworker oder Verbindung geprüft werden. Eine fehlgeschlagene Nachricht benötigt die tatsächliche Fehlerursache statt wiederholter Klicks auf Senden. Bitten Sie Ihre Administration, Postausgang und zugehöriges Ausführungsprotokoll zu prüfen.

Dokumente und Vorlagen

Dokumente aus datensatzbasierten Vorlagen erzeugen und entstandene Datei sowie Anhang prüfen.

Vorlagen und erzeugte Dokumente

Feldwerk kann als Produktfunktion eine Word-Vorlage mit Datensatzwerten füllen und eine DOCX- oder PDF-Datei erzeugen. Das ist von der Ausgabe dieses Handbuchs getrennt: Das Handbuch steht als interaktives HTML bereit, mit PDF als einziger herunterladbarer Ausgabe.

Eine Dokumentvorlage ist ein Datensatz im internen Objekt Dokumentvorlagen. Ihr DOCX-Anhang enthält Layout und Platzhalter. Die Konfiguration bestimmt Quellobjekt, Ausgabeformat, Dateinamen und eine optionale Voraussetzungsformel.

Eine Vorlage vorbereiten

1. Erstellen Sie einen Dokumentvorlagendatensatz mit aussagekräftigem Namen.
2. Setzen Sie das Quellobjekt auf seinen API-Namen; das Beispiel verwendet `chance`.
3. Wählen Sie DOCX- oder PDF-Ausgabe und ein Dateinamensmuster wie `Zusammenfassung-{name}`.
4. Laden Sie die DOCX-Vorlage als Anhang des Vorlagendatensatzes hoch.
5. Konfigurieren Sie eine Dokumenterzeugungsaktion auf der Datensatzseite des Quellobjekts.

Die PDF-Erzeugung benötigt eine konfigurierte und funktionierende LibreOffice-Konvertierung auf dem Server. Die DOCX-Erzeugung füllt die Vorlage ohne diesen Konvertierungsschritt. Ein fehlender Konverter ist ein Serverkonfigurationsproblem und kein Grund, eine DOCX-Datei in PDF umzubenennen.

Ein Vorlagendatensatz, mehrere Dokumente

Ein Vorlagendatensatz kann mehrere DOCX-Anhänge enthalten. Der Generator verarbeitet jede Datei und liefert ein ZIP mit den einzelnen Ausgaben. Das Ausgabeformat gilt für jedes Dokument; der umschließende Download bleibt ein ZIP. Geben Sie Vorlagenanhängen unterschiedliche Dateinamen, damit jede Ausgabe erkennbar bleibt.

Platzhalter einfügen

Ein Content-Platzhalter setzt einen Feldwert ein. Beispielsweise setzt `<# <Content Select="/chance/name"/> #>` den Namen der Verkaufschance ein. Verwenden Sie tatsächliche Feld-API-Namen und unterstützte Pfade. Soweit konfiguriert, hilft das Feldplatzhalter-Widget beim Finden verfügbarer Referenzen.

Ein TableRow-Platzhalter wiederholt eine Tabellenzeile für verwandte untergeordnete Datensätze; Content-Platzhalter darin verwenden relative Pfade wie `./feld`. Prüfen Sie Beziehung und Reihenfolge mit mehreren untergeordneten Testdatensätzen. Unterstützte Bildplatzhalter benötigen einen kompatiblen Dateiwert und Bildtyp.

Die Formatierung folgt den Regeln des Dokumentgenerators, nicht der Oberflächensprache der betrachtenden Person. Der aktuelle Generator formatiert Werte nach deutschen Konventionen. Ein Content-Attribut `Format="zahl"` kann Währungs- oder Prozentzeichen unterdrücken, wenn die Einheit bereits in der Spaltenüberschrift steht.

Eine Tabellenzeile für verwandte Datensätze wiederholen

1. Erstellen Sie eine normale Tabelle in der Word-Vorlage. Halten Sie Spaltenüberschriften in einer separaten Kopfzeile.
2. Fügen Sie in der zu wiederholenden Zeile einen TableRow-Platzhalter für die untergeordnete Beziehung ein. Verwenden Sie die qualifizierte Form `child_object.lookup_field`, wenn mehrere Nachschlagefelder passen könnten.
3. Platzieren Sie relative Content-Platzhalter in den Zellen derselben Zeile. Ein mit `./` beginnender Pfad liest den aktuellen untergeordneten Datensatz.
4. Setzen Sie bei wichtiger Reihenfolge ein ausdrückliches Sort-Attribut und prüfen Sie die Erzeugung mit mindestens zwei untergeordneten Datensätzen.

Beispiel für Quellobjekt `handbuch_dokument` und untergeordnetes Nachschlagefeld `handbuch_zeile.dokument`:

```
<# <TableRow Select="/handbuch_dokument//handbuch_zeile.dokument"
Sort="position asc"/> #>
<# <Content Select="./name"/> #>
<# <Content Select="./betrag" Format="zahl"/> #>
```

Das Beispiel sortiert nach dem untergeordneten Feld `position`, setzt `name` ein und formatiert den Währungsbetrag ohne wiederholtes Währungssymbol. Schreiben Sie „Betrag (EUR)“ in die Kopfzelle, falls dies die Einheit Ihrer Daten ist. Ohne verwandte untergeordnete Datensätze entfällt die Wiederholungszeile; eine normale Kopfzeile bleibt. Der Beziehungslader berücksichtigt Datensatzsichtbarkeit. Prüfen Sie das Ergebnis daher mit dem Konto, das das Dokument erzeugen wird.

Ein Bild aus einem Dateifeld einfügen

Speichern Sie PNG oder JPEG in einem Feld vom Typ **Datei** am Quelldatensatz. Ein allgemeiner Anhang im Dateibereich ist nicht dasselbe wie ein Dateifeldwert. Fügen Sie einen Image-Platzhalter als eigenen Textlauf ein; ein separater Absatz hilft, ihn von umgebendem Text unabhängig zu halten.

```
<# <Image Select="/handbuch_dokument/logo" Width="4cm"/> #>
```

Ersetzen Sie Objekt- und Feldnamen durch Ihre eigenen. Width und Height verwenden Zentimeter. Bei alleiniger Width-Angabe bleibt das Seitenverhältnis erhalten; ohne beide Maße verwendet der Generator 3,5 cm Breite. Die aktuelle Bildauflösung unterstützt Dateifelder direkt am Quelldatensatz, keine mehrstufigen Nachschlagepfade oder Bildfelder untergeordneter Zeilen.

Fehlende oder unlesbare Bilder werden ohne Ersatzmarkierung weggelassen. Prüfen Sie die Ausgabe vor dem Senden, insbesondere bei wesentlichen Bildinformationen. Textplatzhalter können über Word-Textläufe verteilt sein; Image-Platzhalter müssen in einem Textlauf bleiben. Prüfen Sie neben der Vorlage auch die tatsächlich erzeugte PDF.

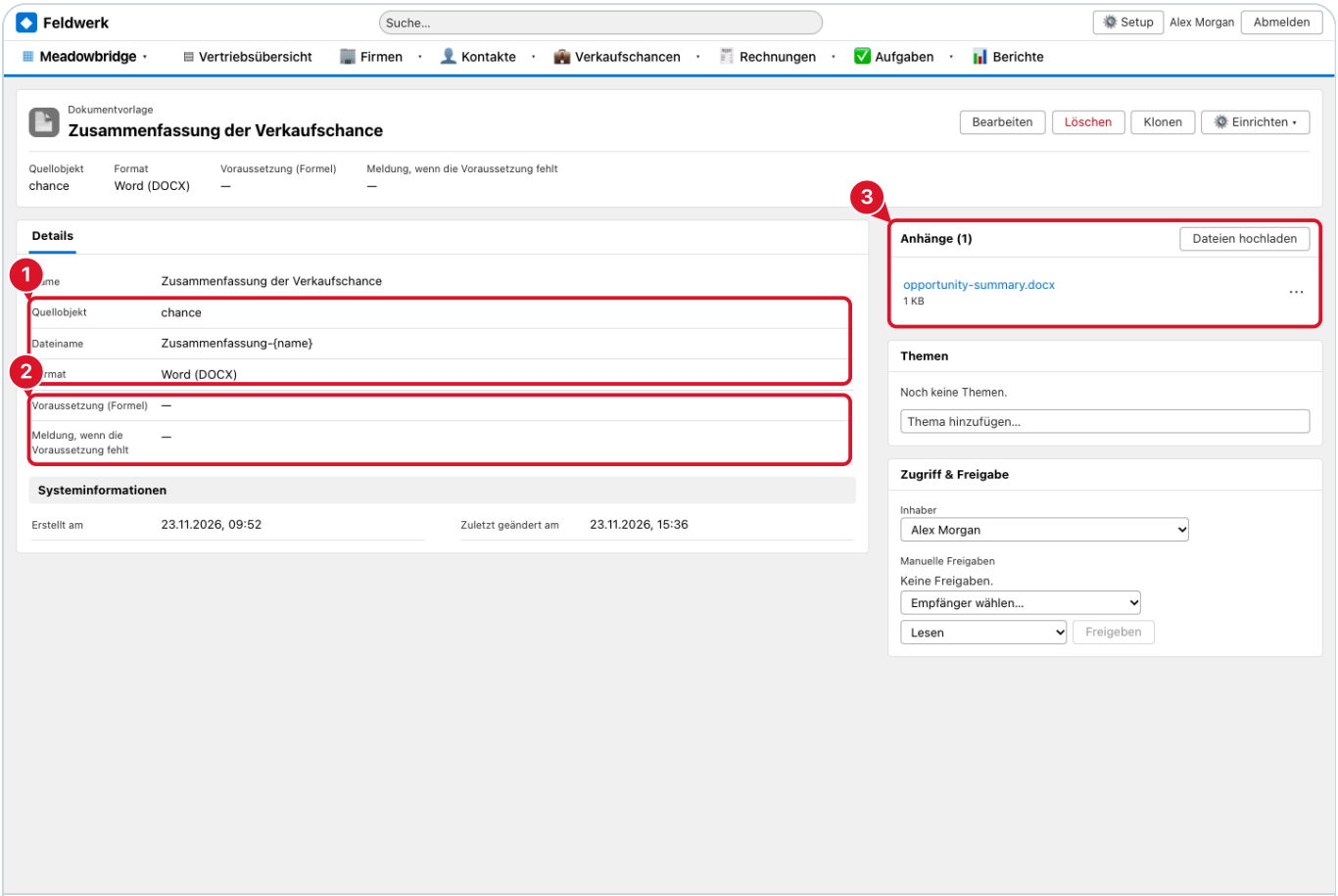
Vollständige Quelldaten verlangen

Eine Vorlagenvoraussetzung ist eine Wahrheitswertformel über die Felder des Quelldatensatzes. Sie muss wahr sein, bevor die Erzeugung erlaubt ist. Das ist die entgegengesetzte Richtung einer Validierungsregel, die den ungültigen Fall beschreibt.

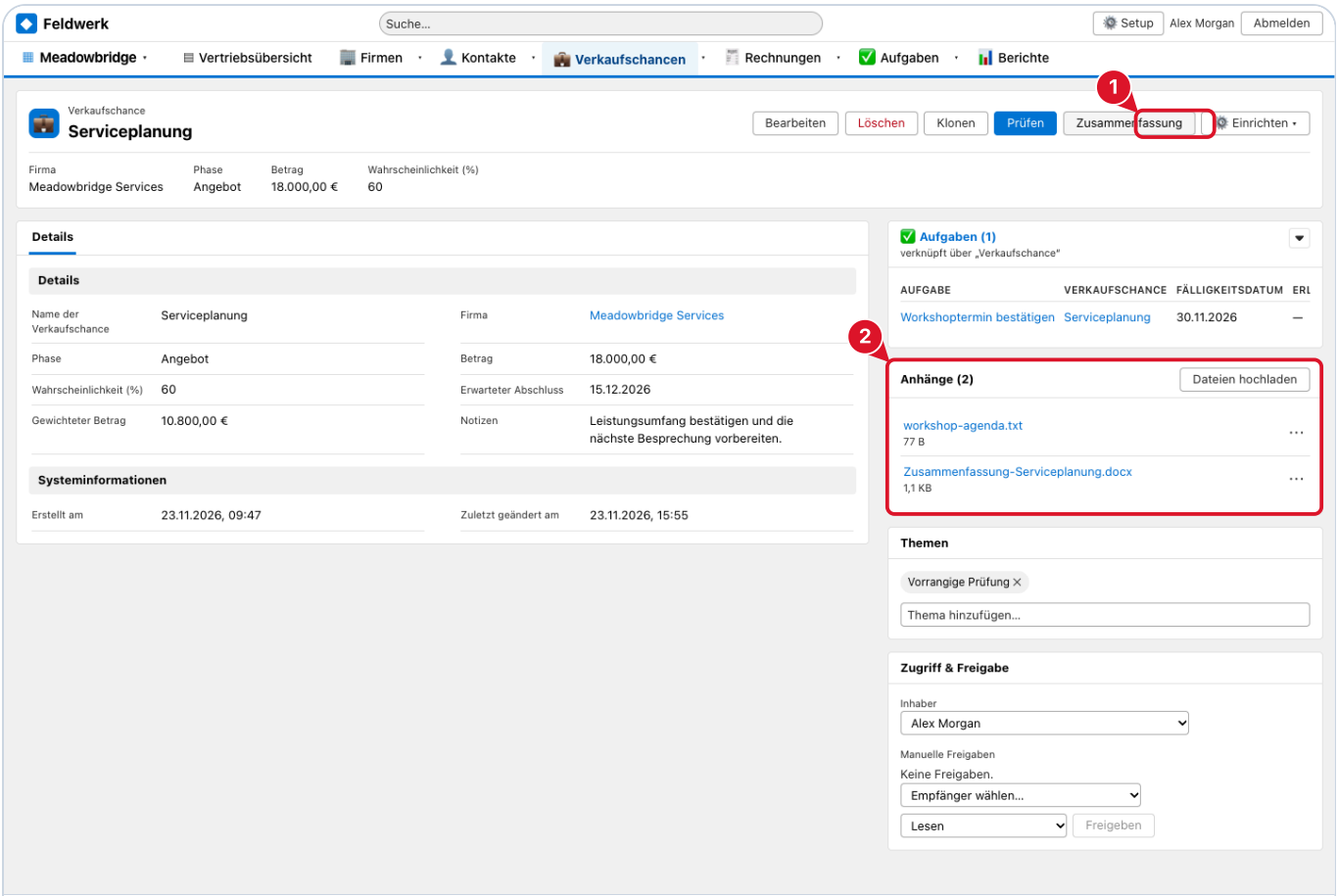
Formulieren Sie eine hilfreiche Voraussetzungsnachricht, etwa „Vergeben Sie vor der Erzeugung dieses Dokuments eine Rechnungsnummer.“ Auch eine fehlerhafte Voraussetzung blockiert die Erzeugung. Die Voraussetzung wird bei Download, Anhangserzeugung und Verwendung als erzeugter E-Mail-Anhang geprüft.

Ausgabe erzeugen und prüfen

1. Öffnen Sie einen Quelldatensatz mit bekannten Werten.
2. Wählen Sie die konfigurierte Dokumentaktion, etwa **Zusammenfassung**.
3. Warten Sie auf den Download oder eine erklärende Fehlermeldung.
4. Öffnen Sie die Datei und prüfen Sie Datensatzname, Werte, Layout und gegebenenfalls wiederholte Zeilen.
5. Prüfen Sie Dateibereich und Ergebnismeldung, um zu erkennen, ob die Ausgabe angehängt wurde.



1. Quellobjekt, Ausgabedateiname und Format wählen.
2. Optionale Voraussetzung und Meldung konfigurieren.
3. Dokumentvorlagendatei anhängen.



1. Konfiguriertes Dokument erzeugen.
2. Entstandenen Anhang und Dateinamen prüfen.

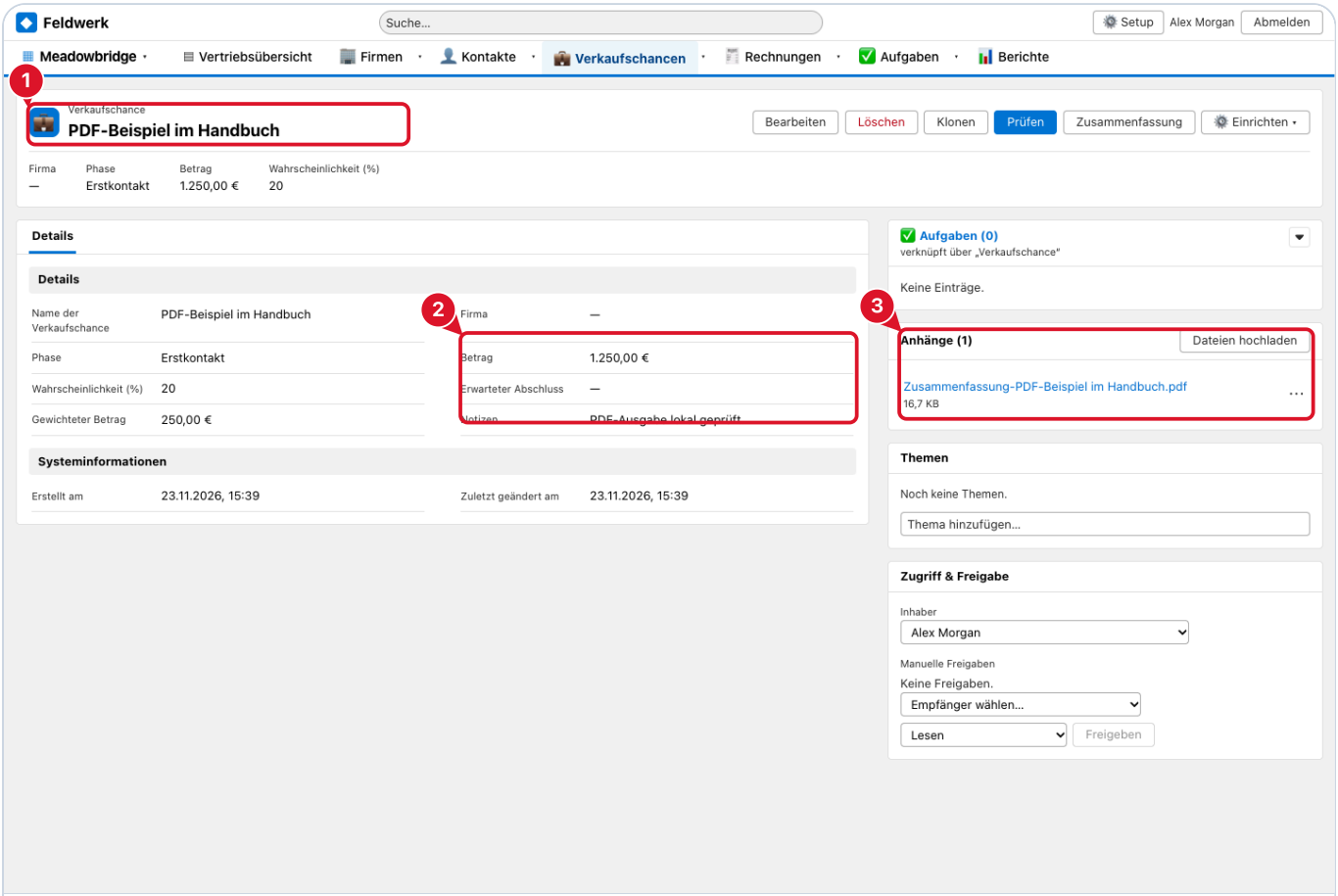
Die Beispielzusammenfassung setzt Serviceplanung und den aktuellen nächsten Schritt in eine DOCX-Datei ein. Ihre Platzhalter werden aufgelöst; die Ausgabe wird an die Verkaufschance angehängt. Eine Wiederholung mit unveränderten Daten erzeugt denselben DOCX-Inhalt und erkennt den identischen Anhang.

Beispiel: eine PDF erzeugen und prüfen

Die geprüfte PDF-Vorlage verwendet Verkaufschancen als Quellobjekt, PDF als Ausgabeformat und `Zusammenfassung-{name}` als Dateinamensmuster. Der Quelldatensatz heißt Handbuch-PDF-Prüfung, mit Betrag 1250 und Notizen „PDF-Ausgabe lokal geprüft."

1. Setzen Sie eine Voraussetzung wie `wert > 2000` und eine hilfreiche Meldung. Die Erzeugung wird blockiert; es entsteht kein Anhang.
2. Ändern Sie die Voraussetzung auf `wert > 1000` und erzeugen Sie erneut.

3. Öffnen Sie die PDF und prüfen Sie, dass Name und Notizen eingesetzt sind und keine Platzhalter übrig bleiben.
4. Bestätigen Sie, dass die PDF unter Dateien am Quelldatensatz erscheint.



1. Quelldatensatz prüfen.
2. Bekannte Quellwerte mit dem erzeugten Dokument vergleichen.
3. PDF-Anhang getrennt vom Download prüfen.

Das Beispiel wurde mit dem lokalen LibreOffice-Konverter umgewandelt und sein PDF-Text geprüft. Vorlagenwerte folgen den Formatierungsregeln des Generators, die von einer englischen Datensatzanzeige abweichen können. In einer separaten Prüfung erzeugten zwei DOCX-Dateien an der Vorlage ein ZIP mit beiden ausgefüllten Dokumenten.

Download und Anhang sind getrennte Ergebnisse

Der Standard-Erzeugungsweg hängt das Dokument normalerweise zusätzlich an, soweit erlaubt. Ein Benutzer mit reinem Lesezugriff kann einen Download ohne Anhangspeicherung erhalten. Lesen Sie das angezeigte Ergebnis, statt beide Vorgänge als erfolgreich vorauszusetzen. Eine konfigurierte Aktion kann die Anhangspeicherung ausdrücklich

verlangen. Im geprüften Lesefall konnte das Mitglied das erzeugte Dokument herunterladen, aber nicht anhängen; verpflichtend angeforderte Anhangspeicherung lieferte einen Fehler. API-Aufrufer verwenden `?ablegen=0` für reinen Download oder `?ablegen=1` für verpflichtende Ablage und prüfen den Antwortheader `X-Anhang-Ablage`.

Die Erkennung identischer Ausgaben bezieht sich auf tatsächlichen Inhalt. Ändern sich Quellwerte oder Vorlageninhalt, kann eine neue Ausgabe gespeichert werden. Leiten Sie Versionsverläufe nicht allein aus Dateinamen ab, da unterschiedliche Dateien gleich heißen können.

Häufige Dokumentprobleme lösen

- **Erzeugung blockiert:** Voraussetzungsnachricht lesen und Quelldatensatz korrigieren.
- **Nicht aufgelöster oder leerer Wert:** Platzhalter, Quellobjekt, Feld-API-Namen, Nachschlagepfad und Zugriff prüfen.
- **Untergeordnete Zeilen fehlen:** Beziehung und Leserecht des erzeugenden Benutzers prüfen.
- **PDF-Umwandlung scheitert:** Administration um Prüfung des Konverters und seiner Protokolle bitten.
- **Download erfolgreich, aber kein Anhang:** Aktionsergebnis und Bearbeitungszugriff prüfen.

Eine elektronische Rechnung erzeugen

Der Generator **XRechnung erzeugen (EN16931 / XRechnung 3.0)** erzeugt eine XML-Datei aus einem Datensatz im Objekt `rechnung` und seinen zugehörigen Positionen. Er wandelt keine normale PDF-Rechnung in eine elektronische Rechnung um. Das angegebene Profil beschreibt das erzeugte XML; erfolgreiche Erzeugung allein belegt nicht, dass ein Empfängersystem es akzeptiert.

Die Administration muss zuerst die vom Generator erwarteten Objekte und Feld-API-Namen bereitstellen. Ein allgemeines Rechnungsobjekt mit nur Name, Betrag und Fälligkeit reicht nicht aus. Anzeigebeschriftungen können unterschiedlich lauten; die folgenden technischen Feldnamen müssen jedoch zum konfigurierten Datenmodell passen.

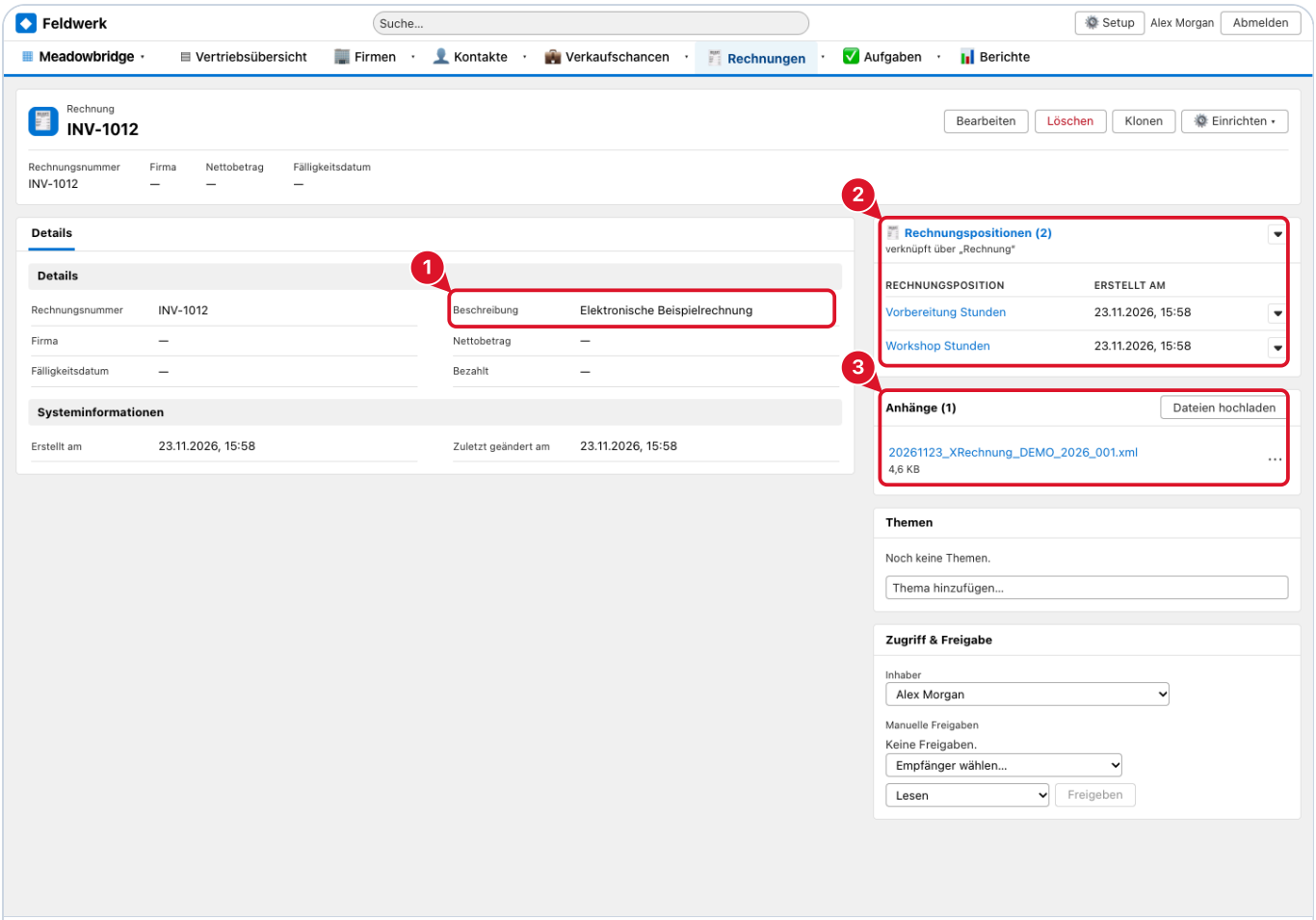
Datensatz	Erforderliche Zuordnung und Werte
Rechnung: `rechnung`	`rechnungsnummer` (ersatzweise `name`), `rechnungsdatum`, `faelligkeit` und `email_primaer` müssen vorhanden sein. Datumswerte verwenden ISO-Daten. Kundendaten stammen aus `rechnungsfirma` sowie `rechnung_strasse`, `rechnung_stadt`, `rechnung_plz` und `rechnung_land`. Der Nettogesamtbetrag stammt aus `summe_netto`.
Lieferant: `firma`	Der Anzeigewert `rechnungssteller` der Rechnung wird mit `firmenname` abgeglichen. Lieferantenadressen verwenden dieselben `rechnung_*`-Feldnamen. Weitere zugeordnete Felder sind `email`, `ust_idnr`, `steuernummer`, `handelsregisternummer`, `rechtsform`, `iban` und `bic`.
Positionen: `belegposten`	Das Nachschlagefeld `rechnung` verbindet jede Position mit ihrer Rechnung. Positionen werden unabhängig vom Anzeigenamen der Rechnung über diese Beziehung geladen. Jede Position verwendet `leistungsbeschreibung`, optional `details`/`leistung_details`, `einheit`, `menge`, `einzelpreis`, `betrag_netto` und `mwst`. Sortiert wird nach Beschreibung.

Da der aktuelle Generator den Lieferanten über Anzeigetext auflöst, setzen Sie das Anzeigefeld des Lieferantenobjekts auf `firmenname`, verwenden Sie einen eindeutigen Lieferantennamen und prüfen Sie den gewählten Lieferanten. Berechtigungen und Feldsichtbarkeit beeinflussen ebenfalls die verfügbaren Generatordaten.

1. Vervollständigen Sie Rechnung, Lieferant und zugehörige Positionen.

2. Öffnen Sie die Rechnung und verwenden Sie die konfigurierte Dokumenterzeugungsaktion für `xrechnung.rechnung`. Administratoren können diesen Generator über die Datensatzseitenaktionen hinzufügen.
3. Laden Sie das XML herunter und prüfen Sie Dateinamen, Rechnungsnummer, Datumswerte, Beteiligte, Zahlungsinformationen, Positionen und Summen.
4. Prüfen Sie die Anhangspeicherung getrennt. API-Benutzer können mit `GET /api/datensaetze/{id}/dokument/xrechnung.rechnung?ablegen=0` ausschließlich einen Download anfordern.
5. Validieren Sie die Datei vor dem Versand gegen das vom Empfängersystem verlangte Profil. Dieser Generator übermittelt sie nicht an ein Rechnungsnetzwerk.

Beispielsweise ergeben Generatortests mit Nettobetrag 1000 eine Steuer von 190 und einen Zahlbetrag von 1190. Der Dateiname folgt `YYYYMMDD_XRechnung_NUMBER.xml`; Zeichen außerhalb von Buchstaben, Ziffern, Unterstrichen und Bindestrichen in der Nummer werden durch Unterstriche ersetzt.



1. Quelldatensatz prüfen.
2. Für die Erzeugung verwendete Rechnungspositionen prüfen.
3. XML-Dateinamen und Anhang kontrollieren.

Das lokale API-Beispiel verwendet zwei zugehörige Positionen mit Nettobeträgen 800 und 200 und speichert den XML-Anhang ausdrücklich. Das allgemeine Seitenlayout zeigt seine vorhandenen Felder; Administratoren können die generatorspezifischen Felder ergänzen. Seine automatische Anzeigenummer ist von der in diesem Beispiel-XML eingesetzten Rechnungsnummer getrennt.

Aktuelle Grenzen des Generators

- Währung EUR, Steuer auf Dokumentebene 19 %, Rechnungstyp 380 und Käuferreferenz `B2B_Rechnung` sind fest eingestellt und werden nicht aus Rechnungsfeldern ausgewählt.
- Die `mwst` einer Position kann abweichen, die Dokumentsumme verwendet dennoch 19 %. Ein fehlender oder auf null gesetzter Positionssteuersatz fällt auf 19 % zurück. Verwenden Sie diese Vorgaben nicht für andere Steuerszenarien.

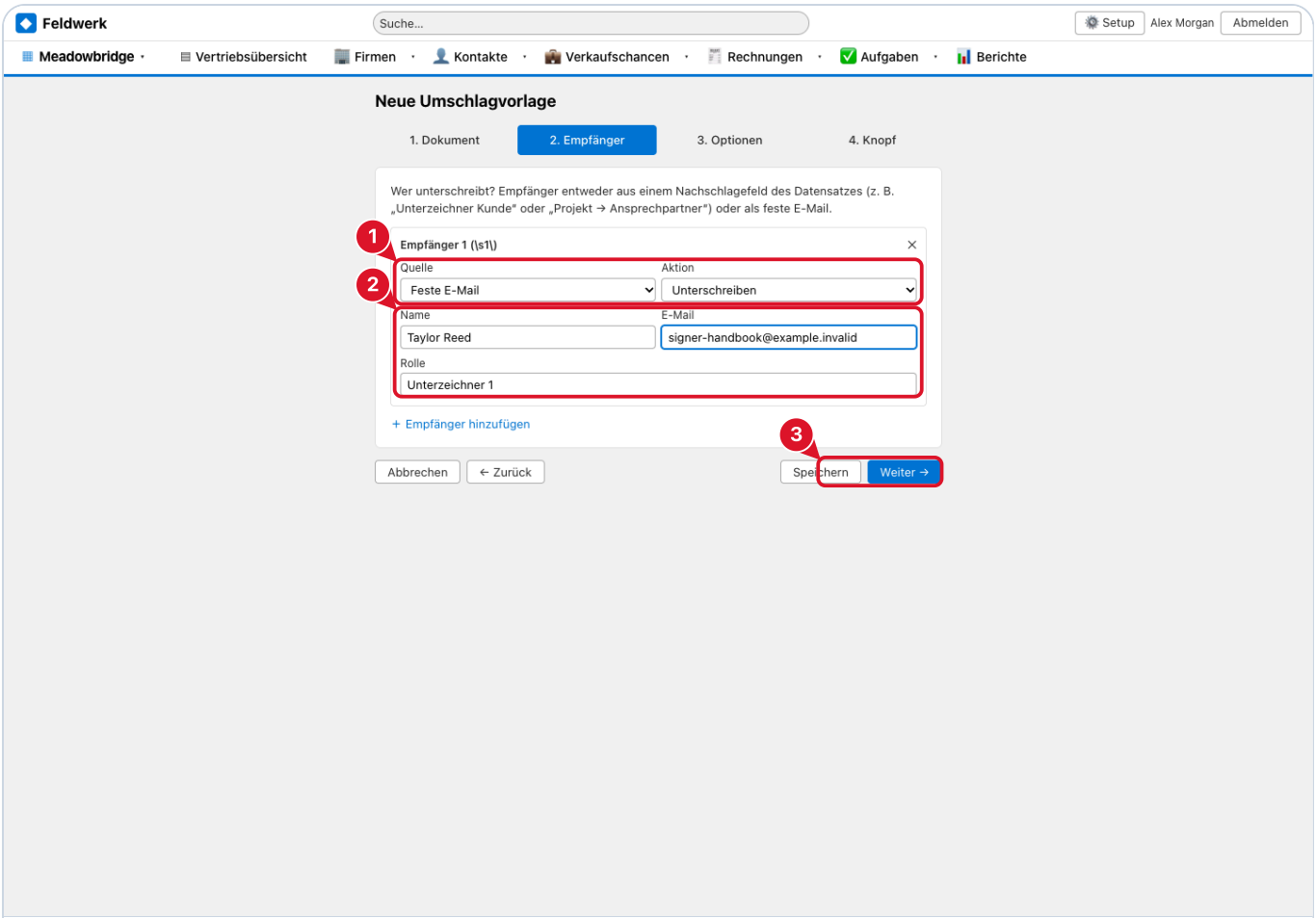
- Der Generator übernimmt `summe_netto` aus der Rechnung und gleicht diesen Betrag nicht mit der Summe der exportierten Positionen ab.
- Der Lieferantenkontaktblock ist nicht implementiert. Fehlende Lieferantendaten können Ersatzwerte statt eines blockierenden Fehlers erzeugen. Prüfen Sie das XML daher auch nach erfolgreicher Erzeugung.
- Einheitenzuordnung verwendet gespeicherte Werte: `Stunden` wird HUR, `Personentage` wird PD und `PT` wird EA. Unbekannte Werte einschließlich unübersetzter englischer Einheitsnamen fallen auf C62 zurück. Ein Oberflächensprachwechsel übersetzt gespeicherte Einheitswerte nicht.
- Erkannte Ländernamen sind Deutschland, Frankreich, Italien, Österreich und Schweiz. Andere Werte werden unverändert durchgereicht. Verwenden Sie geeignete Ländercodes, statt die Übersetzung jedes Ländernamens vorauszusetzen.
- Zahlungsreferenz- und Zahlungsbedingungstexte im XML sind derzeit unabhängig von der Oberflächensprache deutsch.

Meldet die Erzeugung eine fehlende Rechnungsnummer, ein Rechnungsdatum, eine Fälligkeit oder primäre E-Mail, korrigieren Sie den Quelldatensatz und versuchen Sie es erneut. Fehlende Objekte oder Feldzuordnungen erfordern administrative Konfiguration. Die Generatortests des Handbuchs decken XML-Struktur, Feldzuordnung, Summen, Einheiten-/Länderersatzwerte und Fehler für fehlende Felder ab; die Annahme durch das Empfängersystem ist separat zu prüfen.

Eine Umschlagvorlage vorbereiten

Eine Dokumentvorlage liefert die erzeugte Datei; eine Umschlagvorlage liefert den Signaturablauf. Erstellen Sie unter Umschlagvorlagen eine Vorlage und bearbeiten Sie Dokument, Empfänger, Optionen und Schaltfläche.

1. Benennen Sie den Umschlag, wählen Sie sein Quellobjekt und den PDF-Dokumentgenerator.
2. Konfigurieren Sie jeden Empfänger aus einer Datensatzbeziehung oder festen Adresse. Unterscheiden Sie Unterzeichner und Kopieempfänger. Prüfen Sie aufgelösten Namen und Adresse vor dem Senden.
3. Legen Sie E-Mail-Betreff und Nachricht fest. Konfigurieren Sie benötigte Änderungen am Quelldatensatz für Versand, einzelne Unterschriften oder Abschluss.
4. Speichern Sie die Vorlage und konfigurieren Sie ihre Datensatzschaltfläche. Bedingungen können sie auf passende Datensätze einschränken.



1. Empfängerquelle wählen und festlegen, ob die Person unterschreibt.
2. Adresse und Rolle prüfen; Unterzeichnerpositionen entsprechen den Dokumentankern.
3. Vorlage speichern oder zum nächsten Konfigurationsschritt wechseln.

Bei Schaltfläche einrichten im Schritt Schaltfläche werden aktuelle Vorlagenänderungen vor der Datensatzseitenaktion gespeichert. Öffnen Sie einen Quelldatensatz und prüfen Sie die entstandene Schaltfläche Zur Unterschrift senden.

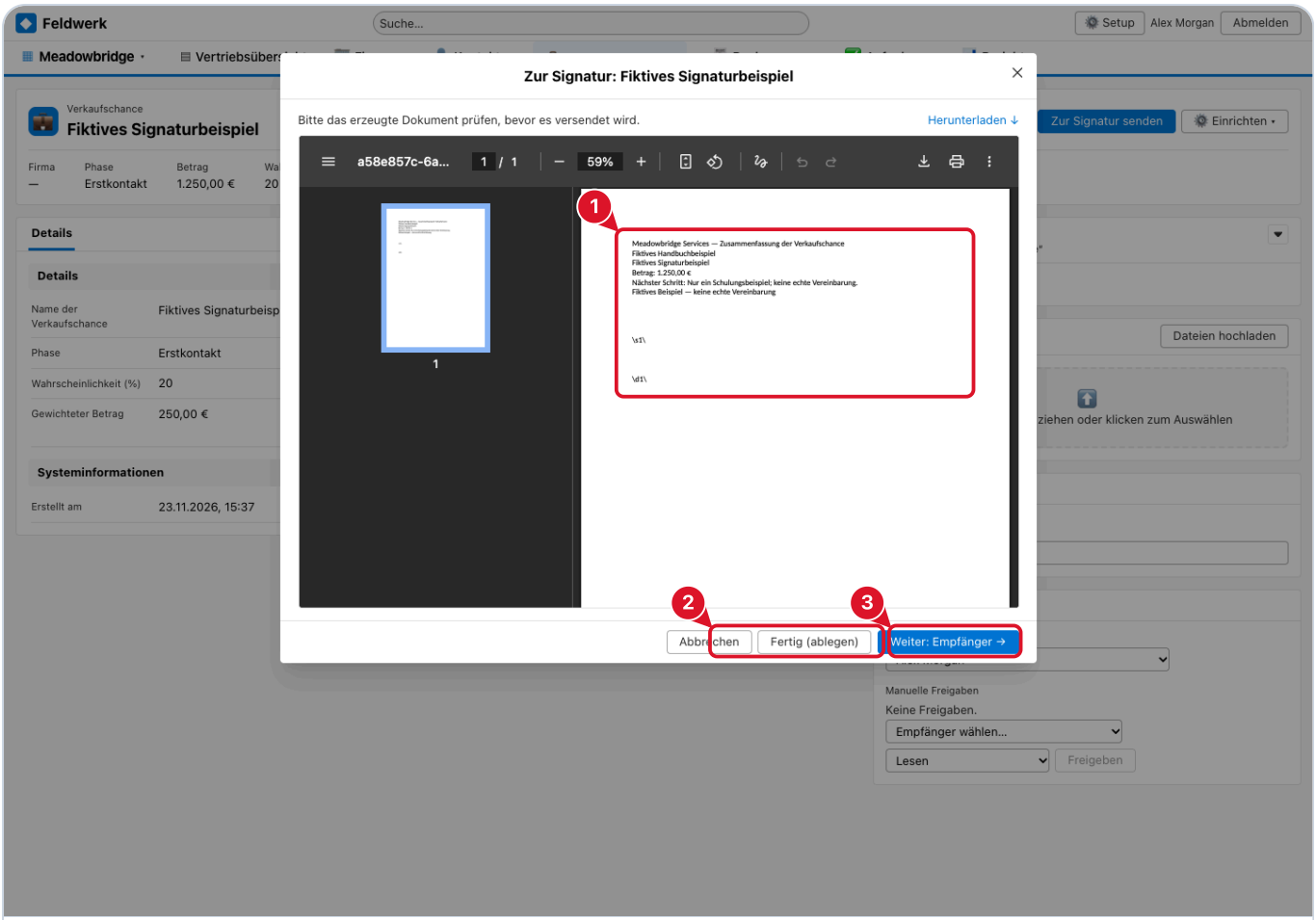
Unterzeichnerpositionen entsprechen den PDF-Ankern `\s1\`, `\s2\` und so weiter. Optionale Datumsanker verwenden `\d1\`, `\d2\`. Kopieempfänger belegen keine Unterzeichnerposition. Prüfen Sie die richtigen Anker im erzeugten PDF vor dem Versand: Eine normale PDF ohne Unterschriftsfelder bietet im öffentlichen Betrachter keinen Ort zum Unterschreiben.

Die aktuelle Versandaktion verwendet parallele Weiterleitung. Kopieempfänger erhalten die abgeschlossenen Dokumente ohne Signierlink. Erinnerungs- und Ablaufwerte einer Vorlage sind Konfigurationsfelder; die aktuelle Versandaktion richtet nicht automatisch Erinnerungen ein und übernimmt die Ablauffrist nicht in den Auftrag. Konfigurieren und testen Sie die gewünschte Automatisierung separat.

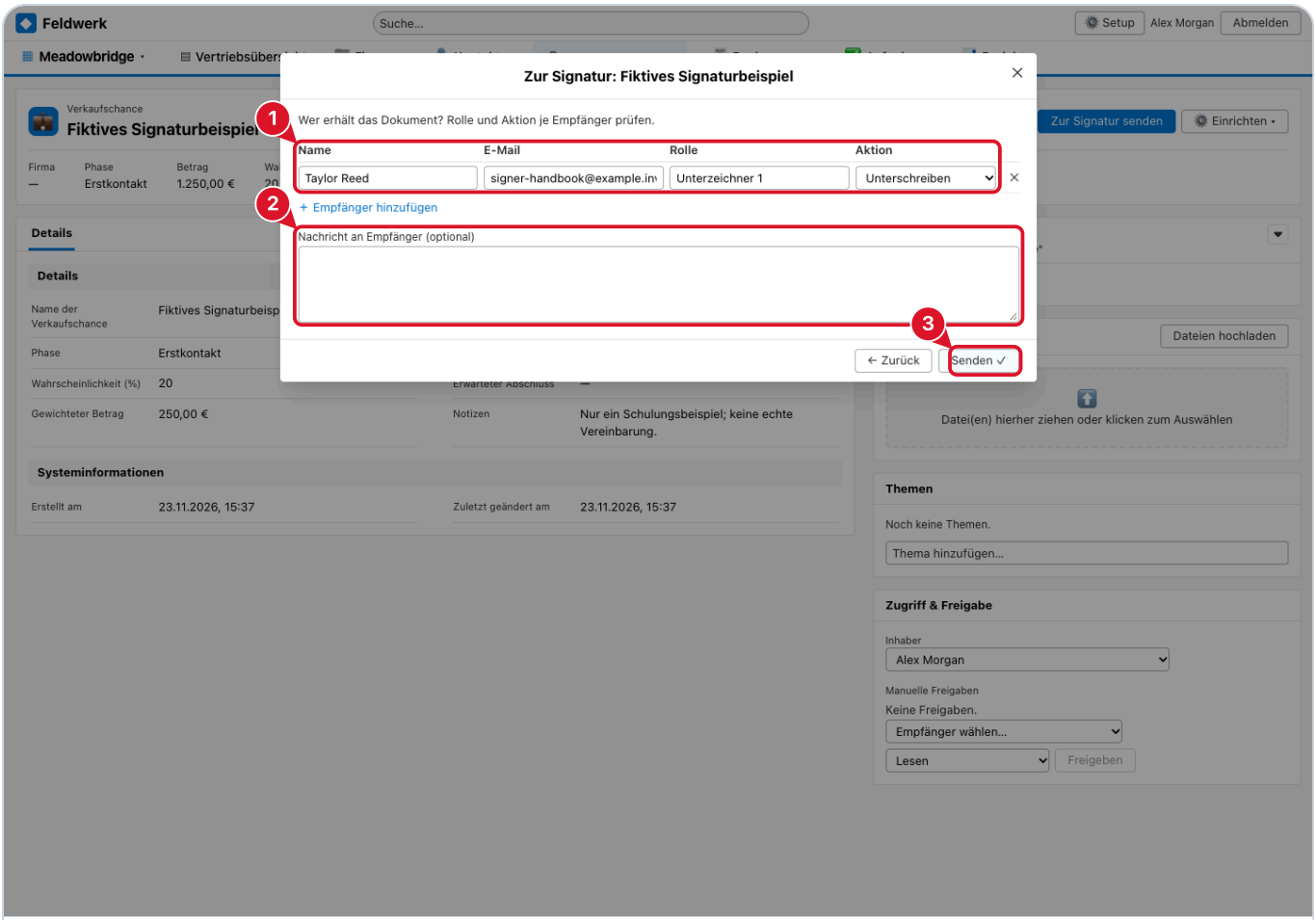
Eine Signaturanfrage prüfen und senden

Die Datensatzaktion **Zur Unterschrift senden** kann den Signaturassistenten öffnen. Prüfen Sie erzeugtes Dokument und aufgelöste Empfänger vor der Bestätigung. Die Vorbereitung löst Konfiguration auf, ohne eine Anfrage zu senden. Das Senden erstellt einen Signaturauftrag, speichert Original-PDF und Hash, legt Unterzeichnerdatensätze an und reiht Einladungs-E-Mails ein.

1. Öffnen Sie den Quelldatensatz und wählen Sie **Zur Unterschrift senden**.
2. Prüfen Sie PDF, Werte und Signaturanker. Das Öffnen dieser Vorschau oder Herunterladen hängt keine Datei an den Quelldatensatz an.
3. Wählen Sie **Fertig (als Anhang speichern)**, um das Dokument ohne Signaturanfrage abzulegen. Mit **Weiter: Empfänger** fahren Sie zur Unterschrift fort.
4. Prüfen Sie jeden Namen, jede E-Mail-Adresse, Rolle und Aktion. Ein Unterzeichner benötigt die Aktion Unterschreiben; ein Kopieempfänger unterschreibt nicht. Ergänzen oder korrigieren Sie Empfänger bei Bedarf.
5. Geben Sie optional eine Nachricht ein und wählen Sie **Senden**. Mit Zurück prüfen Sie das Dokument erneut.



1. Erzeugte Werte und Signaturanker prüfen.
2. Als Anhang speichern, ohne eine Anfrage zu senden.
3. Vor dem Senden zur Empfängerprüfung fortfahren.



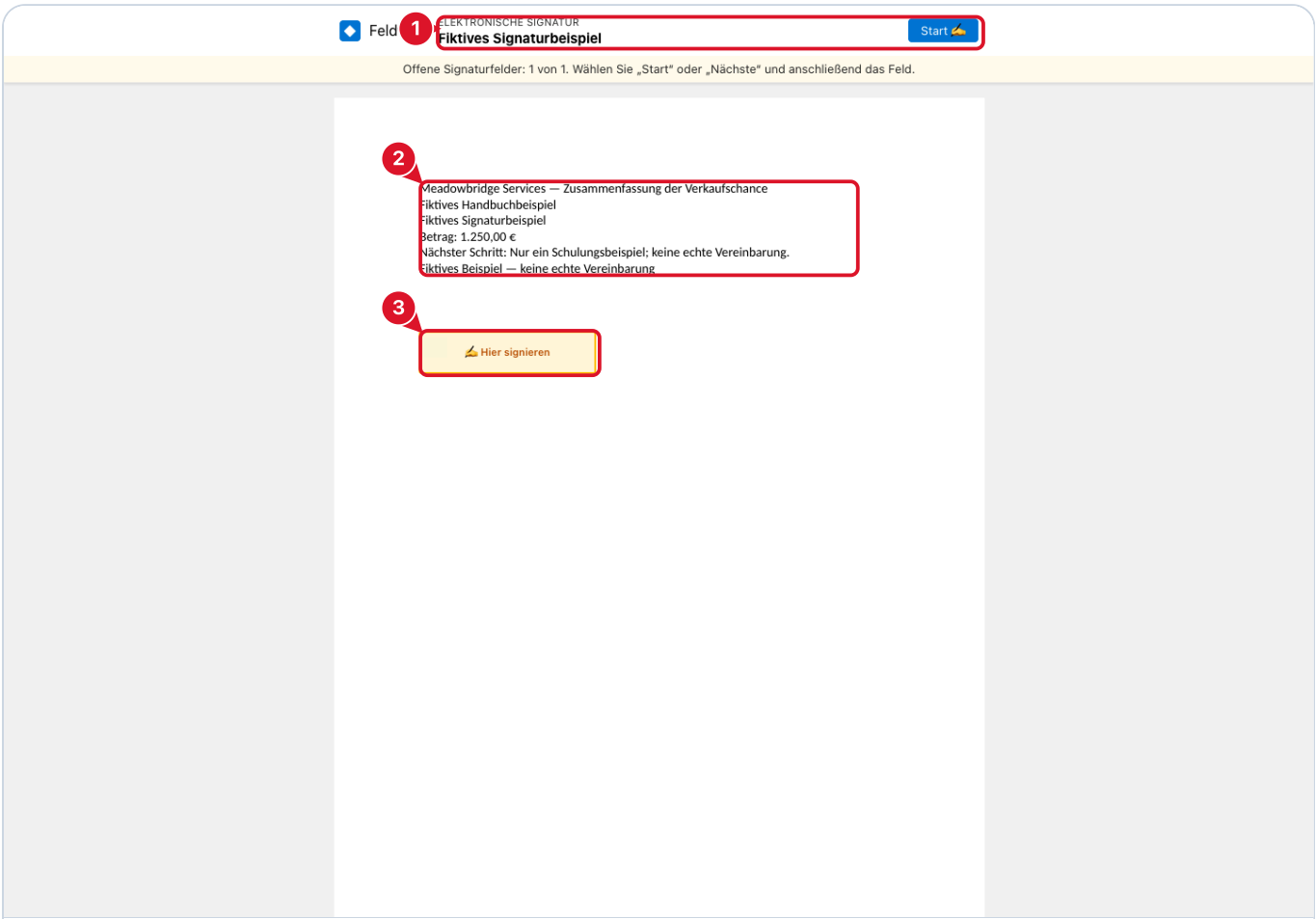
1. Jeden Namen, jede Adresse, Rolle und Aktion prüfen.
2. Bei Bedarf eine Nachricht ergänzen.
3. Senden reiht Einladungen ein; Zustellung separat im Postausgang prüfen.

Halten Sie den Signierlink jedes Empfängers vertraulich. Er gewährt ohne Feldwerk-Konto Zugriff auf dessen Dokument. Prüfen Sie Auftrag und Postausgang getrennt: Eine eingereihte Einladung beweist keinen Empfang. Die Handbuchdemonstration verwendet eine lokale Warteschlange und fiktive Adressen; sie versendet keine externe E-Mail.

Als Empfänger unterschreiben

1. Öffnen Sie Ihren persönlichen Signierlink und lesen Sie das Dokument.
2. Wählen Sie Start, um zum ersten Unterschriftsfeld zu gelangen, und anschließend Hier unterschreiben.
3. Geben Sie Ihren Namen ein oder wechseln Sie zu Zeichnen. Prüfen Sie die angezeigte Unterschrift.

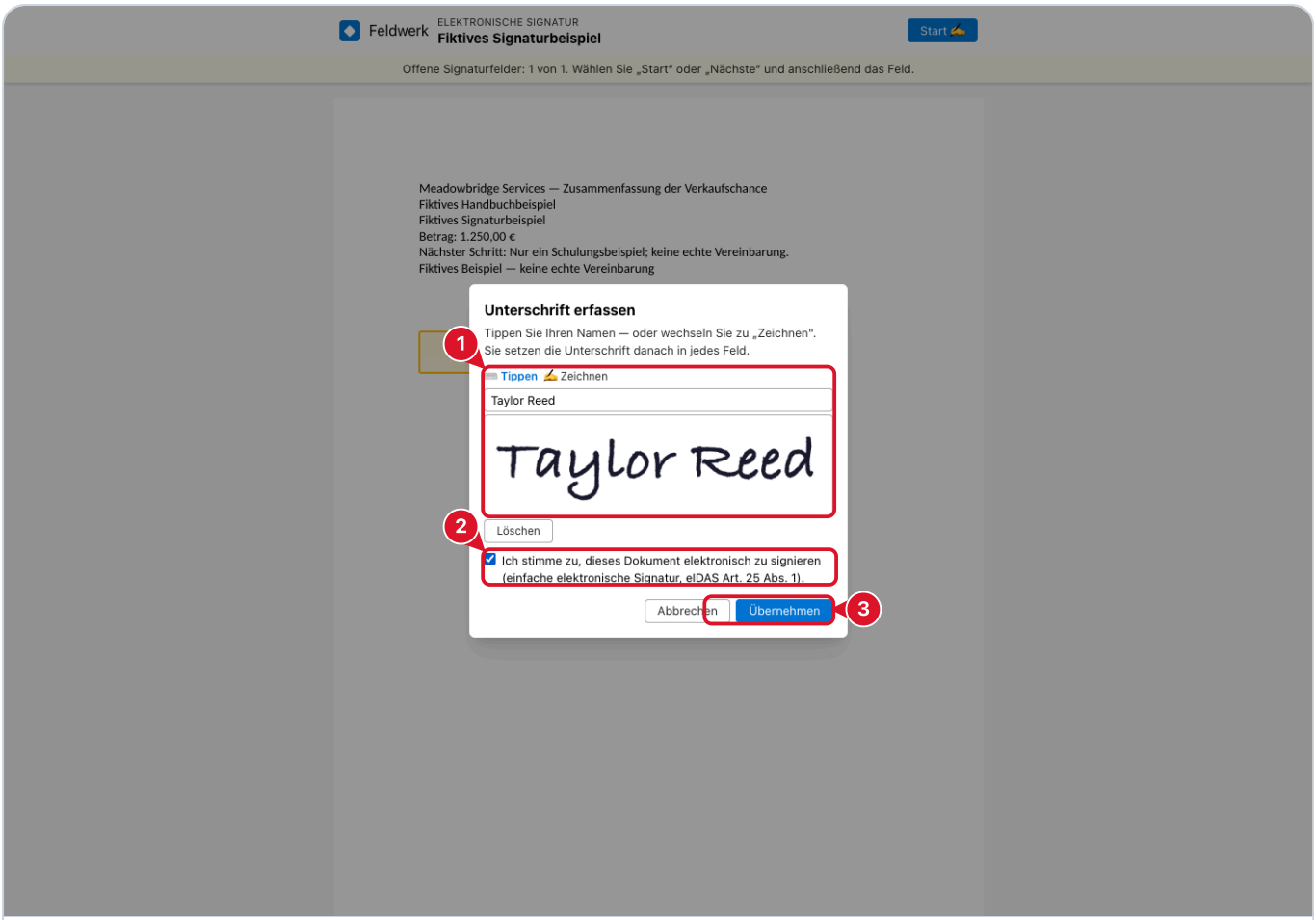
4. Lesen und aktivieren Sie die Einwilligungscheckbox. Übernehmen bleibt ohne Unterschrift und Einwilligung gesperrt.
5. Wählen Sie Übernehmen. Bei weiteren Feldern wählen Sie jedes verbleibende Feld, um dieselbe Unterschrift einzusetzen.



1. Auftragstitel prüfen und mit Start das eigene Feld finden.

2. Dokument vor dem Unterschreiben lesen.

3. Markiertes Unterschriftsfeld wählen.

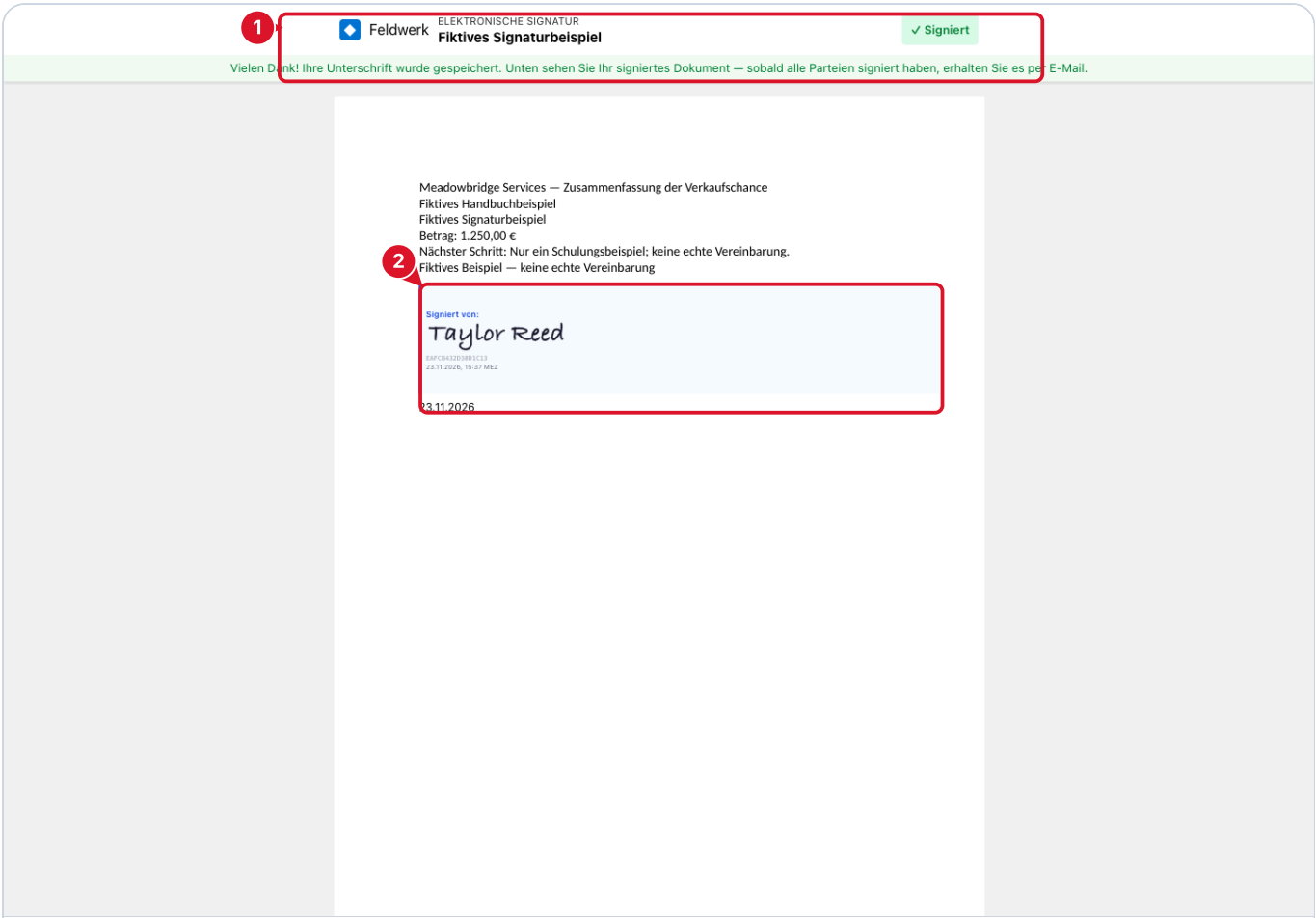


1. Unterschrift tippen oder zeichnen und Vorschau prüfen.
2. Einwilligung lesen und ausdrücklich ankreuzen.
3. Unterschrift übernehmen oder Dialog abbrechen.

Scrollen Sie auf einem schmalen Bildschirm innerhalb des PDF-Betrachters horizontal, um die ganze Seite zu prüfen. Start bringt Ihr Unterschriftsfeld in die Ansicht.

Das Ausfüllen des letzten zugewiesenen Unterschriftsfelds übermittelt die Unterschrift nach kurzer Verzögerung automatisch. Es gibt keinen zusätzlich zu suchenden abschließenden Bestätigungsbildschirm. Scheitert die Übermittlung, lesen Sie den Fehler und prüfen Sie zunächst, ob die Unterschrift erfasst wurde. Verwenden Sie danach Abschließen für einen erneuten Versuch.

Abschluss und entstandene Dateien prüfen



1. Status Unterschrieben und Bestätigung prüfen.
2. Sichtbare Unterschrift und Datum prüfen; weitere Empfänger müssen gegebenenfalls noch unterschreiben.

Der Status Unterschrieben bestätigt die Erfassung der Unterschrift dieses Empfängers. Er bedeutet nicht, dass bereits alle anderen Empfänger unterschrieben haben. Beim erneuten Öffnen desselben Links erscheint eine Bestätigung statt einer erneuten Unterschriftsanforderung.

Nach den Unterschriften aller Unterzeichner wird der Auftrag als abgeschlossen markiert. Feldwerk versucht anschließend, das unterschriebene PDF, ein separates Abschlusszertifikat und Abschluss-E-Mails zu erzeugen. Diese Nachverarbeitung erfolgt nach bestem Bemühen: Prüfen Sie Dateien des Auftrags und Postausgang, statt sich allein auf den Abschlussstatus zu verlassen. Das geprüfte fiktive Beispiel erzeugte Original-PDF, unterschriebenes PDF und Zertifikat.

Das Zertifikat protokolliert Signaturereignisse und Dokumentidentifikation. Betrachten Sie die erzeugten Dokumente als Ergebnis dieses konfigurierten Ablaufs; dieses Handbuch belegt keine Eignung für eine bestimmte gesetzliche Formvorgabe.

Eine Anfrage erneut senden oder stornieren

Ist eine Einladung nicht angekommen, prüfen Sie zuerst den Postausgang. Verwenden Sie die konfigurierte Erneut-senden-Aktion des Auftrags für eine Ersatzeinladung. Sie ersetzt das Signiertoken des Empfängers; der bisherige Link funktioniert danach nicht mehr. Bitten Sie den Empfänger, die neueste Einladung zu verwenden. Dafür ist weder ein zweiter Auftrag noch ein anderes Dokument nötig.

Stoppen Sie eine Anfrage über ihre konfigurierte Stornoaktion und prüfen Sie den stornierten Status. Ein stornierter Auftrag lehnt weitere Unterschriftsübermittlungen ab, auch wenn ein Empfänger den alten Link noch geöffnet hat. Ungültige, abgelaufene oder stornierte Links sind mit dem Absender zu klären. Die lokale Demonstration prüfte sowohl die Ungültigkeit ersetzter Links als auch die Ablehnung einer Unterschrift nach Stornierung.

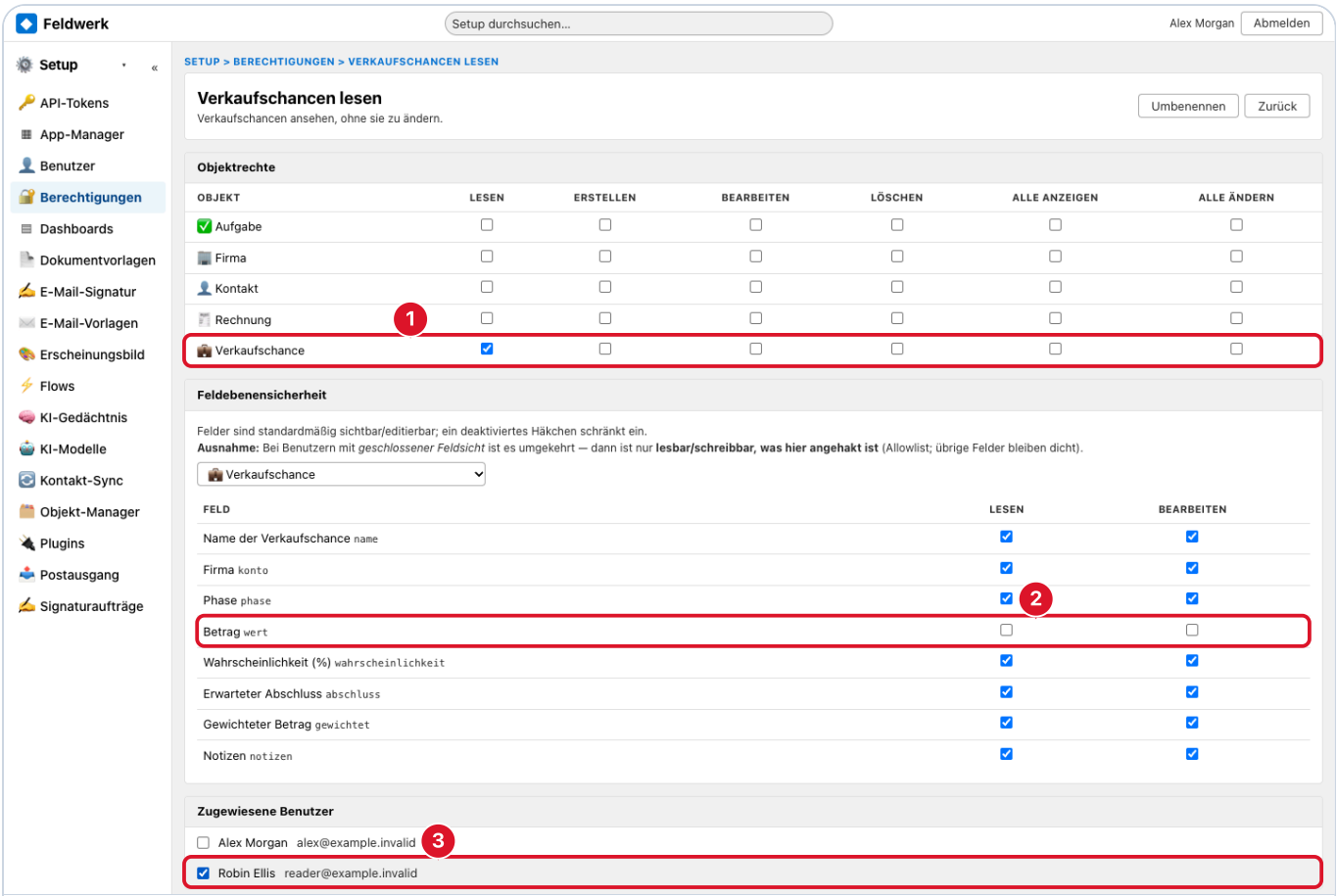
Benutzer, Berechtigungen und Freigaben

Menschen den benötigten Zugriff geben und ihn mit ihrem eigenen Konto prüfen.

Die Zugriffsebenen verstehen

Feldwerk kombiniert Objektberechtigungen, Feldberechtigungen und Datensatzfreigaben. Eine Objektberechtigung bestimmt, ob eine Person Datensätze dieses Typs lesen, erstellen, bearbeiten oder löschen darf. Feldberechtigungen steuern einzelne Werte. Freigaben bestimmen, auf welche Datensätze eine Person zugreifen kann. Ein sichtbares Register allein gewährt keines dieser Rechte.

Administratoren haben vollständigen Zugriff. Prüfen Sie Einschränkungen mit einem Mitgliedskonto: Ein Test als Administrator kann keinen Feldschutz belegen. Ein Mitglied kann mehrere Berechtigungssätze besitzen; die Erlaubnisse der Zuweisungen werden kombiniert. Ein einschränkender Satz hebt eine durch einen anderen Satz erteilte Berechtigung nicht auf.



1. Lesen für Verkaufschance gewähren, ohne Bearbeiten zu erlauben.
2. Lesen und Bearbeiten für Betrag einschränken.
3. Den Satz Robin Ellis zuweisen.

Einen Berechtigungssatz erstellen

1. Öffnen Sie **Setup → Berechtigungen** und wählen Sie **Neuer Berechtigungssatz**.
2. Vergeben Sie einen klaren Namen, etwa Verkaufschancen lesen, und beschreiben Sie den Zweck.
3. Öffnen Sie den Satz und erteilen Sie die benötigten Objektberechtigungen. Beginnen Sie mit Lesen für Verkaufschancen, wenn die Rolle nur die Pipeline prüfen soll.
4. Prüfen Sie die Feldberechtigungen dieses Objekts. Blenden Sie beispielsweise Betrag aus, wenn die Rolle die Verkaufschance ohne ihren wirtschaftlichen Wert sehen soll.
5. Weisen Sie den Satz den vorgesehenen Benutzern zu und prüfen Sie deren übrige Zuweisungen.
6. Melden Sie sich als Mitglied an und testen Sie den resultierenden Zugriff.

Alle anzeigen und **Alle ändern** sind weitreichende objektbezogene Ausnahmen von der Datensatzfreigabe. Sie unterscheiden sich von normalem Lesen und Bearbeiten. Vergeben Sie sie nur an Rollen, die auf alle Datensätze dieses Objekts zugreifen müssen.

Den Feldzugriff sorgfältig prüfen

Im normalen Feldzugriffsmodus sind Felder innerhalb des gewährten Objektzugriffs standardmäßig verfügbar, sofern sie nicht eingeschränkt werden. Ein anderer zugewiesener Satz kann das Feld wieder freigeben. Bei eng begrenzten Integrationskonten ändert **Geschlossener Feldzugriff** in den Benutzereinstellungen den Standard: Nur ausdrücklich freigegebene Felder sind verfügbar, auch wenn Administratoren später neue Felder hinzufügen.

Ein berechnetes Feld hat eigene Feldberechtigungen. Das Verbergen eines Eingangswerts verbirgt nicht automatisch eine daraus abgeleitete Formel. Prüfen Sie neben dem ursprünglichen Feld auch abhängige Formel- und Aggregatergebnisse. Anzeigenamen in Nachschlagefeldern verdienen ebenfalls eine separate Prüfung: Ein in einer Beziehung sichtbarer Name ist nicht gleichbedeutend mit der Berechtigung, den verknüpften Datensatz zu öffnen.

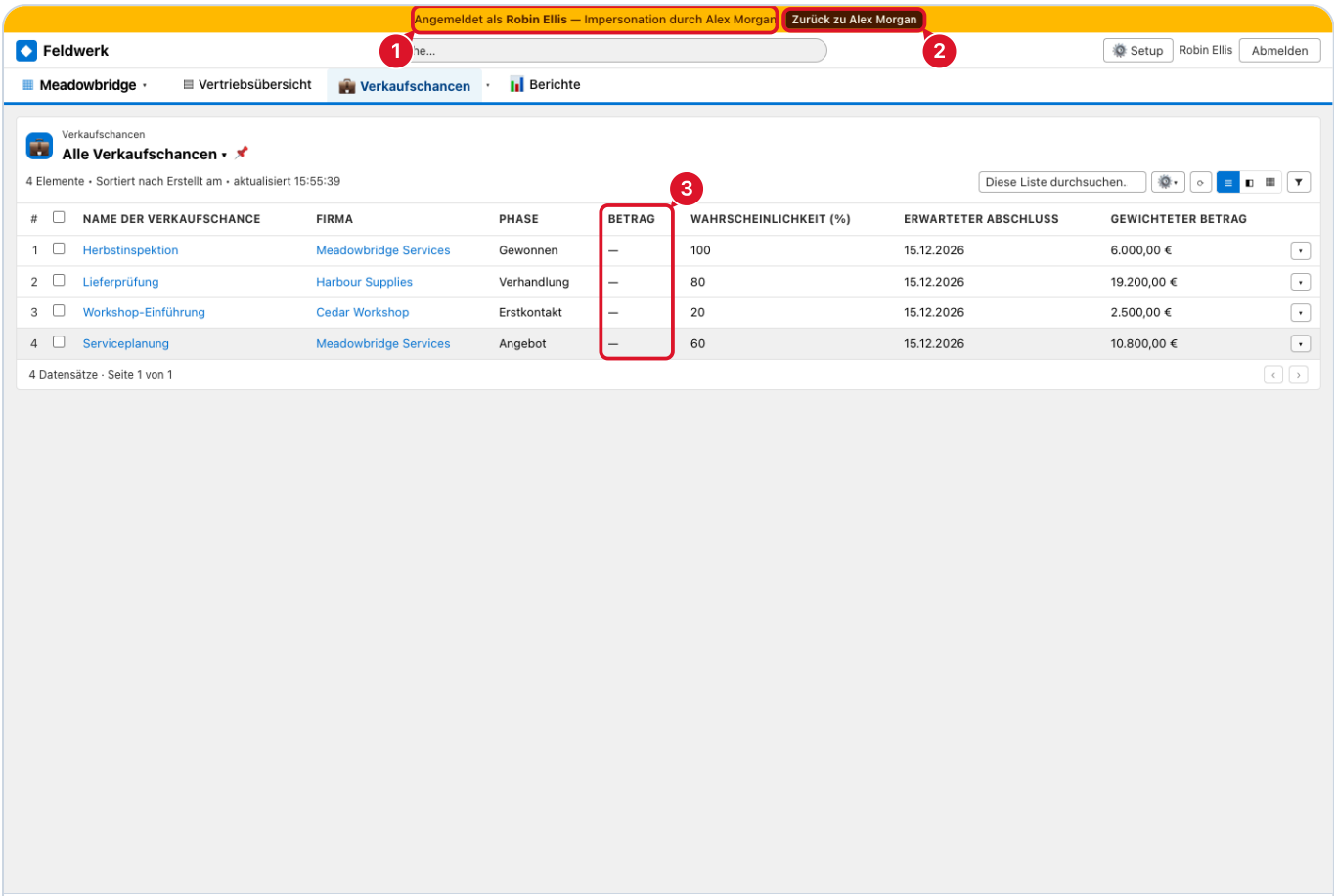
Einzelne Datensätze steuern

Die Standardfreigabe eines Objekts kann privat, öffentlich lesbar oder öffentlich les- und schreibbar sein. Eigentümerschaft, manuelle und kriterienbasierte Freigaben können Zugriff ergänzen. Objektberechtigungen bleiben erforderlich. Wer einen freigegebenen Datensatz lesen kann, erhält nicht automatisch dessen Bearbeitungsrecht.

1. Wählen Sie im Objekt-Manager die Standardfreigabe des Objekts.
2. Vergeben Sie für jeden Datensatz einen geeigneten Eigentümer.
3. Ergänzen Sie eine Freigaberegel, wenn ein wiederholbares Kriterium Zugriff für einen Benutzer oder Berechtigungssatz gewähren soll.
4. Verwenden Sie für Ausnahmen eine einzelne Datensatzfreigabe und wählen Sie bewusst Lesen oder Lesen/Schreiben.
5. Testen Sie Listen, einzelne Datensätze, verwandte Listen, Suche und Berichte erneut mit dem betroffenen Konto.

Benutzer und Supportzugriff verwalten

Unter **Setup → Benutzer** verwalten Sie Konten und deren Rolle als Administrator oder Mitglied. Administratoren können mit der Aktion zur Anmeldung als Benutzer die Ansicht eines aktiven Kontos prüfen. Die Sitzung verwendet dessen Berechtigungen; ein sichtbarer Hinweis bietet die Rückkehraktion. Kehren Sie nach Abschluss zur Administratorsitzung zurück. Eine Ansicht zu prüfen ist kein Grund, dem Benutzer Administratorrechte zu geben.



1. Prüfen, wessen Berechtigungen aktiv sind.
2. Zum Administratorkonto zurückkehren.
3. Eingeschränkte Betragswerte sind verborgen; berechnete Felder haben eigene Berechtigungen.

Prüfbeispiel: Verkaufschancen nur lesen

Im Handbuchbeispiel erhält Robin Ellis Leserecht auf Verkaufschancen und darf Betrag nicht lesen. Robin kann Serviceplanung abrufen; das Feld `wert` fehlt jedoch. Eine Namensänderung und die Verwaltung von Berechtigungssätzen werden abgelehnt. Auch

derselbe Lesezugriff über einen angemeldeten MCP-Client lässt Betrag aus. Alex Morgan behält als Administrator vollständigen Zugriff.

Wiederholen Sie diese Prüfung nach Änderungen an Zuweisungen. Testen Sie bei Integrationen auch das Token des Kontos, da ein Administratortoken Fehler in der Mitgliedskonfiguration verdecken würde. Siehe API und MCP.

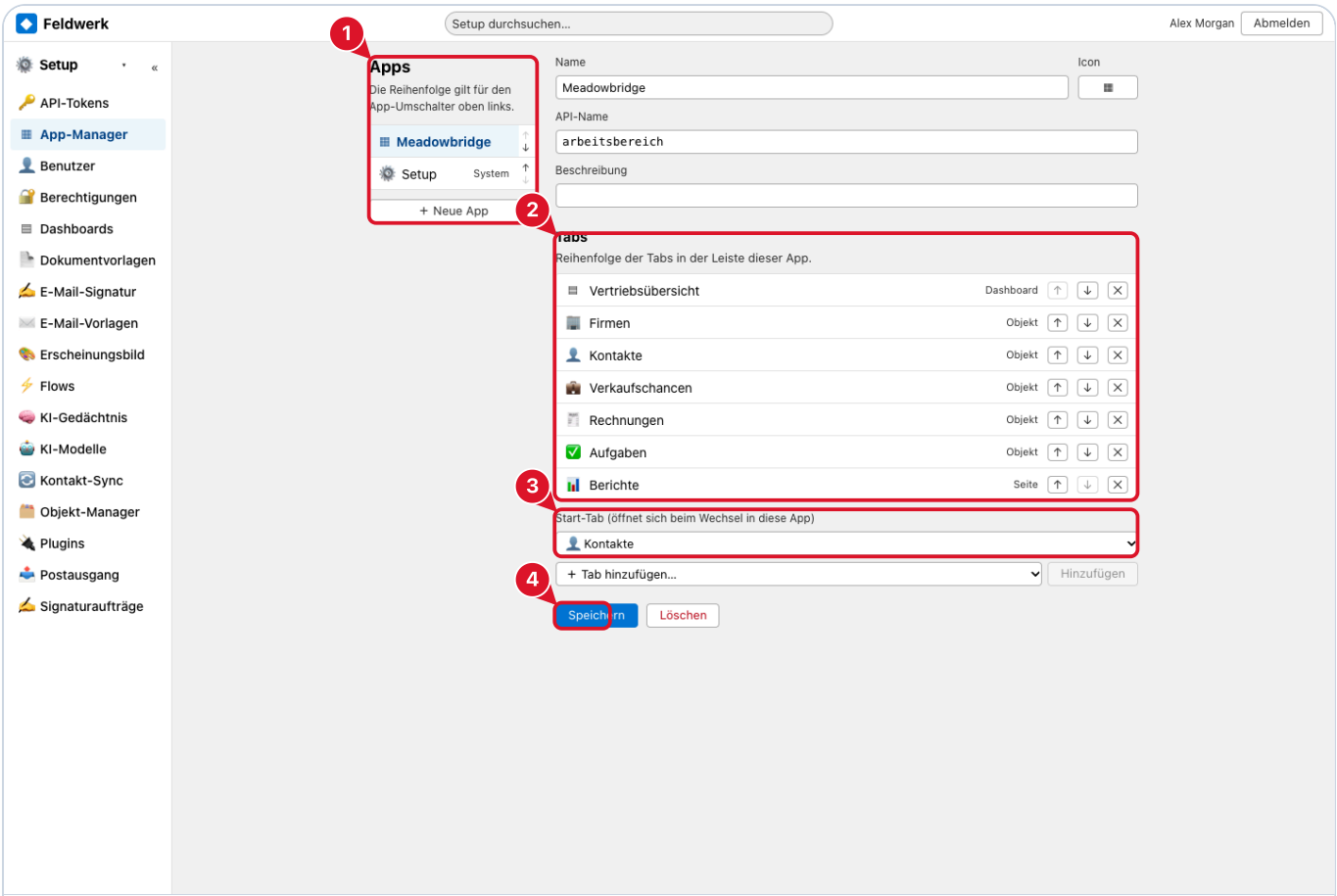
Apps und Erscheinungsbild

Navigation für ein Team zusammenstellen und dem Arbeitsbereich sein Erscheinungsbild geben.

Den Arbeitsbereich für die Aufgabe wählen

Eine App fasst Navigationsregister zu einem Arbeitsbereich zusammen. Sie kann Objektlisten, Dashboards und unterstützte Seiten enthalten. Sie kopiert keine Datensätze und erstellt keine separate Datenbank. Dasselbe Kontaktobjekt kann in mehreren Apps erscheinen; der Zugriff richtet sich weiterhin nach den Benutzerberechtigungen.

Wechseln Sie Apps über die App-Auswahl oben links. Beim Wechsel öffnet sich ein konfiguriertes Startregister. Im Beispiel Meadowbridge ist Kontakte das Startregister; daneben stehen Vertriebsübersicht, Firmen, Verkaufschancen, Rechnungen, Aufgaben und Berichte zur Verfügung.



1. App wählen oder erstellen.
2. Register anordnen.
3. Startregister wählen.
4. App speichern.

Eine App erstellen oder ändern

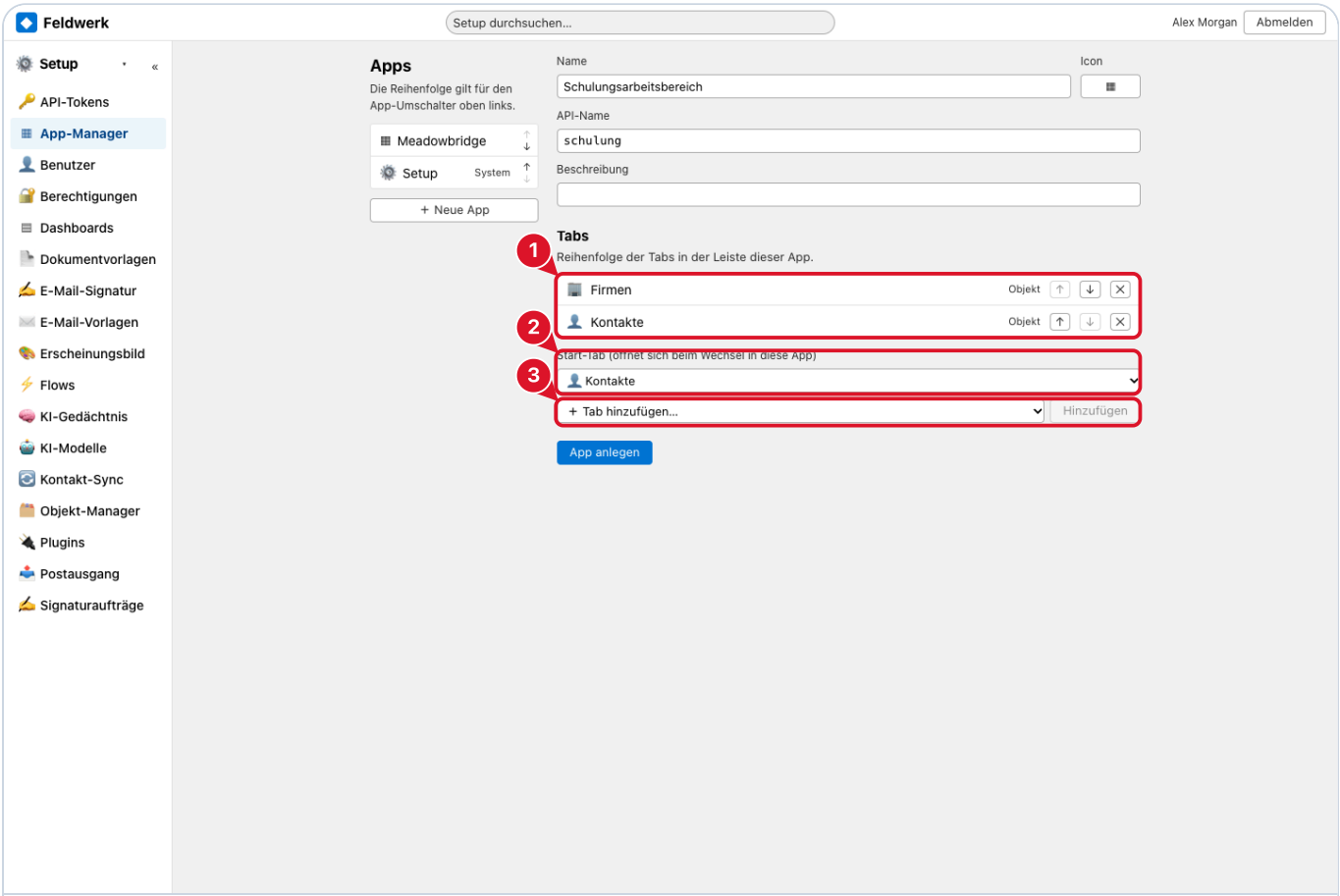
1. Öffnen Sie **Setup → App-Manager**.
2. Wählen Sie eine vorhandene App oder **Neue App**.
3. Geben Sie Name, Symbol, API-Namen und Beschreibung ein. Der Name sollte dem Team den Zweck der App erklären.
4. Wählen Sie über **Register hinzufügen…** ein verfügbares Objekt, Dashboard oder eine Seite und anschließend **Hinzufügen**.
5. Verschieben Sie Register mit den Auf-/Ab-Schaltflächen. Entfernen Sie ein Register über seine Entfernen-Aktion, wenn es nicht mehr in dieser App erscheinen soll.
6. Wählen Sie das **Startregister**. Ohne ausdrückliches Ziel ist das erste Objektregister der

Standard.

7. Wählen Sie bei einer neuen App **App erstellen**, bei einer vorhandenen **Speichern**. Wechseln Sie zur App und prüfen Sie die Navigation.

Die Pfeile in der App-Liste ändern die Reihenfolge der Apps in der Auswahl; die Pfeile neben Registern ändern die Navigation innerhalb der gewählten App. Das Entfernen eines Registers löscht weder sein Objekt noch dessen Datensätze. Das Löschen einer App entfernt ihre Navigationskonfiguration; prüfen Sie zuvor, wer sie verwendet. Die Setup-App ist eine System-App.

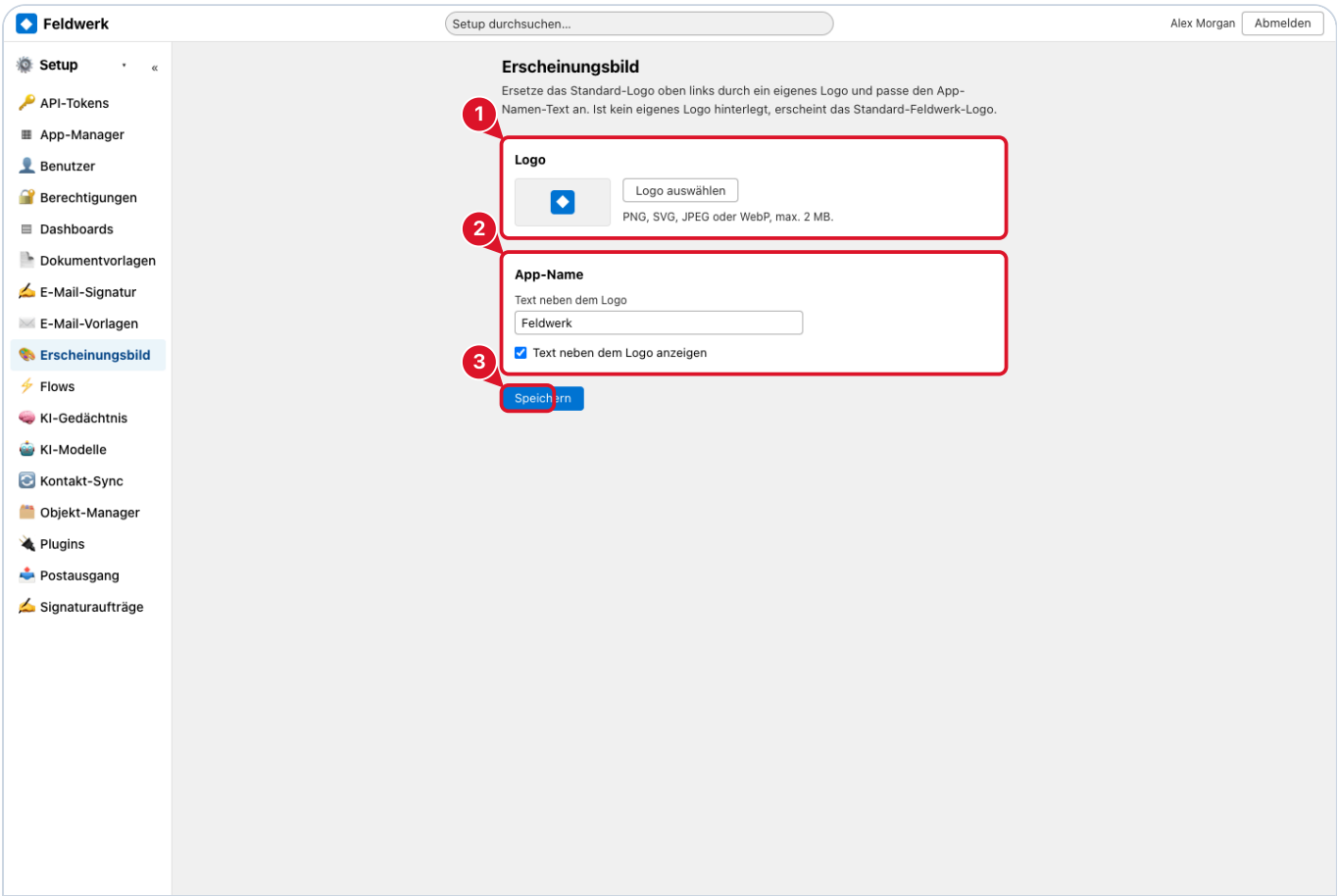
In der folgenden Konfiguration stehen Firmen vor Kontakte, aber Kontakte ist ausdrücklich als Startregister gewählt. Der App-Wechsel öffnet daher Kontakte, obwohl Firmen zuerst erscheint.



1. Register anordnen und unerwünschte Einträge entfernen.
2. Ziel beim App-Wechsel wählen.
3. Ein weiteres Objekt, Dashboard oder eine unterstützte Seite hinzufügen.

Ein hilfreiches Startdashboard hinzufügen

Erstellen und speichern Sie das Dashboard, bevor Sie es als Register hinzufügen. Für ein Vertriebsteam kann die Vertriebsübersicht einen Pipelinebericht und eine Aufgabenliste zeigen. Wählen Sie sie als Startregister, wenn Benutzer zuerst die Übersicht sehen sollen. Dashboardwidgets verwenden weiterhin den Zugriff des aktuellen Benutzers; eine Administratorvorschau kann mehr zeigen als die Ansicht eines Mitglieds. Siehe Dashboards.



1. Logo hochladen oder entfernen.
2. Name und Sichtbarkeit festlegen.
3. Texteinstellungen speichern.

Logo und Namen ändern

1. Öffnen Sie als Administrator **Setup → Erscheinungsbild**.
2. Wählen Sie **Logo auswählen** und eine PNG-, SVG-, JPEG- oder WebP-Datei bis 2 MB. Der Upload wird getrennt von den Texteinstellungen angewendet.
3. Prüfen Sie Logovorschau und Kopfzeile.

4. Geben Sie den neben dem Logo anzuzeigenden Text ein und legen Sie fest, ob er sichtbar sein soll.
5. Wählen Sie **Speichern**, um die Texteinstellungen anzuwenden.

Mit **Logo entfernen** kehren Sie zum Standardlogo zurück. Ohne eigenes Logo verwendet Feldwerk seine Standardmarke. Das Erscheinungsbild gilt für die gesamte Instanz, nicht nur für die aktuell ausgewählte App.

Das Ergebnis prüfen

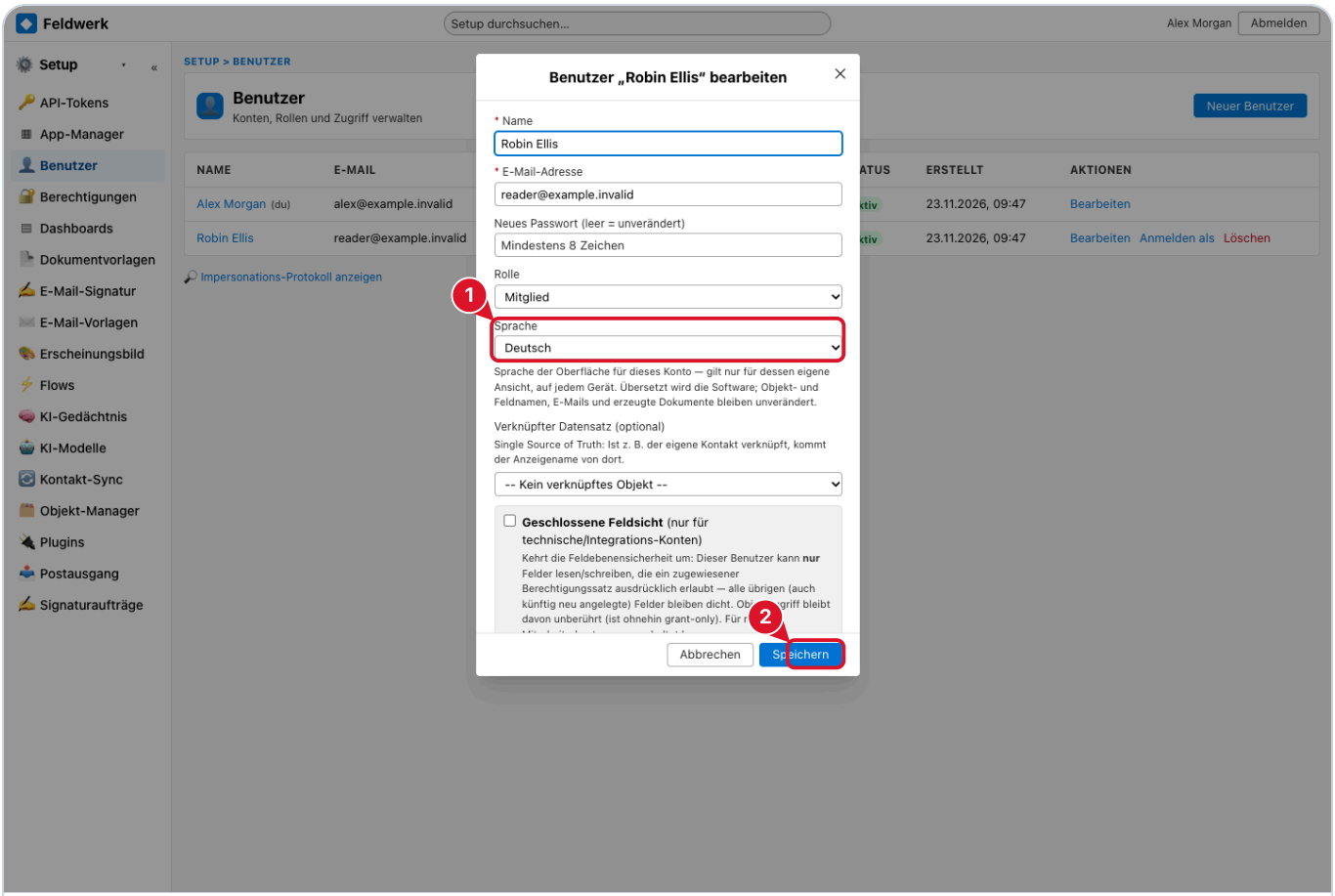
Testen Sie App-Auswahl, Startregister und jeden Navigationseintrag mit dem Konto eines Teammitglieds. Fehlt ein Objekt, prüfen Sie sowohl sein App-Register als auch seine Zugriffsrechte. Ist ein Dashboard leer, prüfen Sie Widgets und Datenberechtigungen, bevor Sie die App ändern. Prüfen Sie das Logo auch bei schmaler Browserbreite, nicht nur auf dem Desktop.

Die Oberflächensprache eines Kontos wählen

Administratoren können unter **Setup → Benutzer** das gewünschte Konto bearbeiten, die **Sprache** wählen und speichern. Die Einstellung gehört zu diesem Konto; Kollegen können daher unterschiedliche Sprachen in derselben Instanz verwenden. Für Ihr eigenes Konto akzeptiert die API-Operation `PUT /api/auth/sprache` die Werte `{"sprache":"en"}` oder `{"sprache":"de"}`; MCP bietet ebenfalls Sprachwerkzeuge.

Die Oberflächensprache ändert Plattformbedienelemente und unterstützte Systemmetadaten. Objektnamen, Feldbeschriftungen, eigene Validierungsmeldungen, Formularinhalte und Datensatzwerte Ihrer Organisation bleiben wie konfiguriert. Übersetzen Sie diese gezielt für ihre Zielgruppe, statt zu erwarten, dass ein Oberflächensprachwechsel sie umschreibt. Öffentliche Besucher ohne angemeldetes Konto verwenden ihre Browsersprache für unterstützte Plattformtexte.

Sprachnamen erscheinen in ihrer eigenen Sprache, etwa **Deutsch** und **English**. So bleiben sie auch bei unbekannter aktueller Oberflächensprache erkennbar.



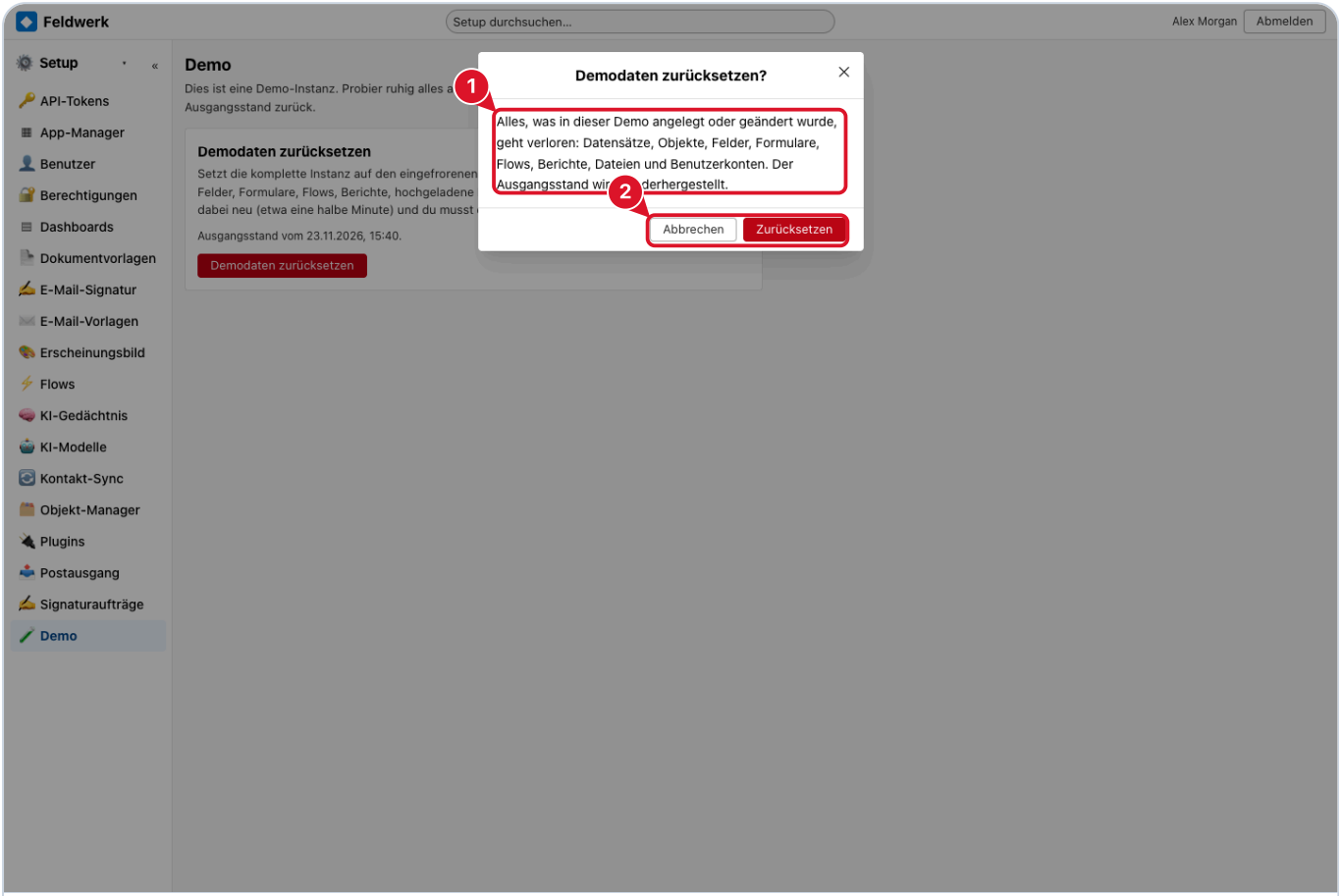
1. Oberflächensprache für dieses Konto wählen.
2. Speichern und anschließend die eigene Sitzung des Benutzers prüfen.

Eine eigene Demoinstanz zurücksetzen

Setup → Demo betrifft nur eine ausdrücklich für Demonstrationen konfigurierte Instanz. Eine normale Instanz zeigt **Diese Instanz läuft nicht im Demo-Modus.** Dies ist kein Weg, eine einzelne Fehlbearbeitung rückgängig zu machen.

1. Prüfen Sie, dass Sie in der vorgesehenen Demoinstanz sind und den angezeigten Ausgangsschnappschuss wiederherstellen möchten.
2. Wählen Sie **Demodaten zurücksetzen**.
3. Lesen Sie die Bestätigung: Seit dem Schnappschuss erstellte oder geänderte Datensätze, Objekte, Felder, Formulare, Flows, Berichte, hochgeladene Dateien und Benutzerkonten gehen verloren.
4. Wählen Sie **Abbrechen**, um den aktuellen Stand zu behalten, oder **Zurücksetzen**, um den Schnappschuss wiederherzustellen.

5. Warten Sie auf den Serverneustart und melden Sie sich danach mit einem Konto aus dem wiederhergestellten Schnappschuss an.



1. Vor dem Fortfahren den Umfang des Resets prüfen.
2. Abbrechen erhält die aktuelle Demo; Zurücksetzen stellt ihren Schnappschuss wieder her.

Ohne verfügbaren Ausgangsschnappschuss ist die Zurücksetzen-Schaltfläche deaktiviert. Wird der Neustart nicht abgeschlossen, folgen Sie dem angezeigten Neuladehinweis und kontaktieren Sie die Instanzadministration. Ein Demoreset hat nichts mit dem Löschen einer App oder dem Entfernen eines Navigationsregisters zu tun.

KI-Assistent und Gedächtnis

Den Assistenten mit Datensatzkontext nutzen, vorgeschlagene Änderungen prüfen und gemerkte Fakten verwalten.

Bevor Sie beginnen

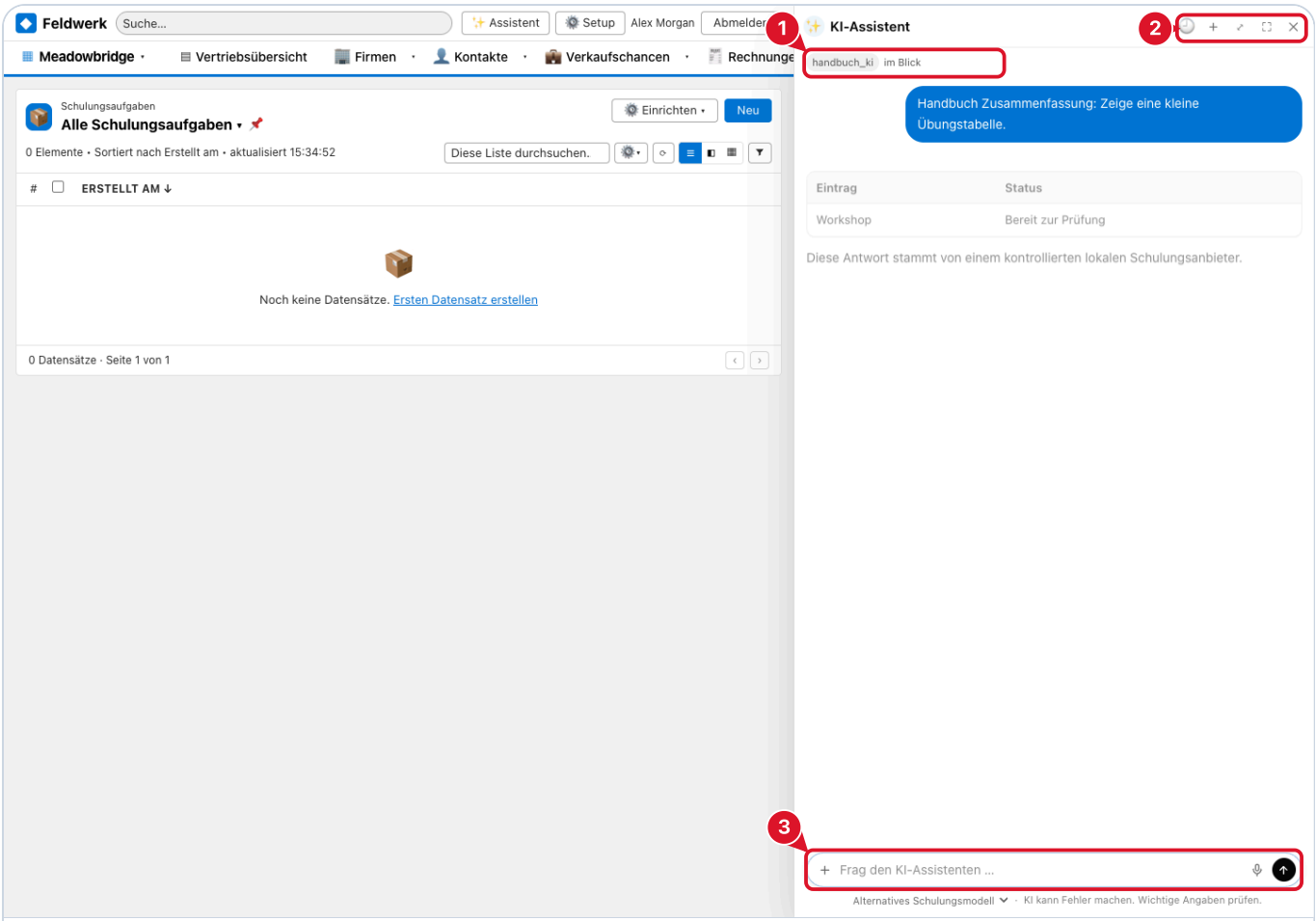
Der KI-Assistent benötigt ein konfiguriertes verfügbares Modell. Administratoren verwalten Modelldatensätze unter **Setup → KI-Modelle**. Ist der Assistent nicht verfügbar, bitten Sie die Administration, Modell- und Anbieterkonfiguration zu prüfen. Ohne konfigurierten Anbieter lässt sich ein Gespräch nicht allein durch Öffnen des Bereichs führen.

Der Assistent verwendet dieselben Anwendungswerkzeuge und Berechtigungsprüfungen wie unterstützte API- und MCP-Operationen. Er kann Datensätze finden, Daten erklären und Änderungen vorbereiten. Seine Antwort muss dennoch geprüft werden, insbesondere vor dem Versenden von Informationen oder dem Anwenden von Änderungen.

Mit klarem Kontext fragen

1. Öffnen Sie das zu besprechende Objekt oder den Datensatz und wählen Sie oben **Assistent**.
2. Prüfen Sie den im Bereich angezeigten Objekt- oder Datensatzkontext.
3. Stellen Sie eine konkrete Frage, etwa „Fasse die Verkaufschance Serviceplanung zusammen und liste die in ihren Notizen genannten nächsten Schritte auf.“
4. Prüfen Sie die Werkzeugergebnisse und öffnen Sie den referenzierten Datensatz, um wichtige Fakten zu kontrollieren.
5. Fragen Sie nach, wenn Umfang, Zeitraum oder benötigte Felder fehlen.

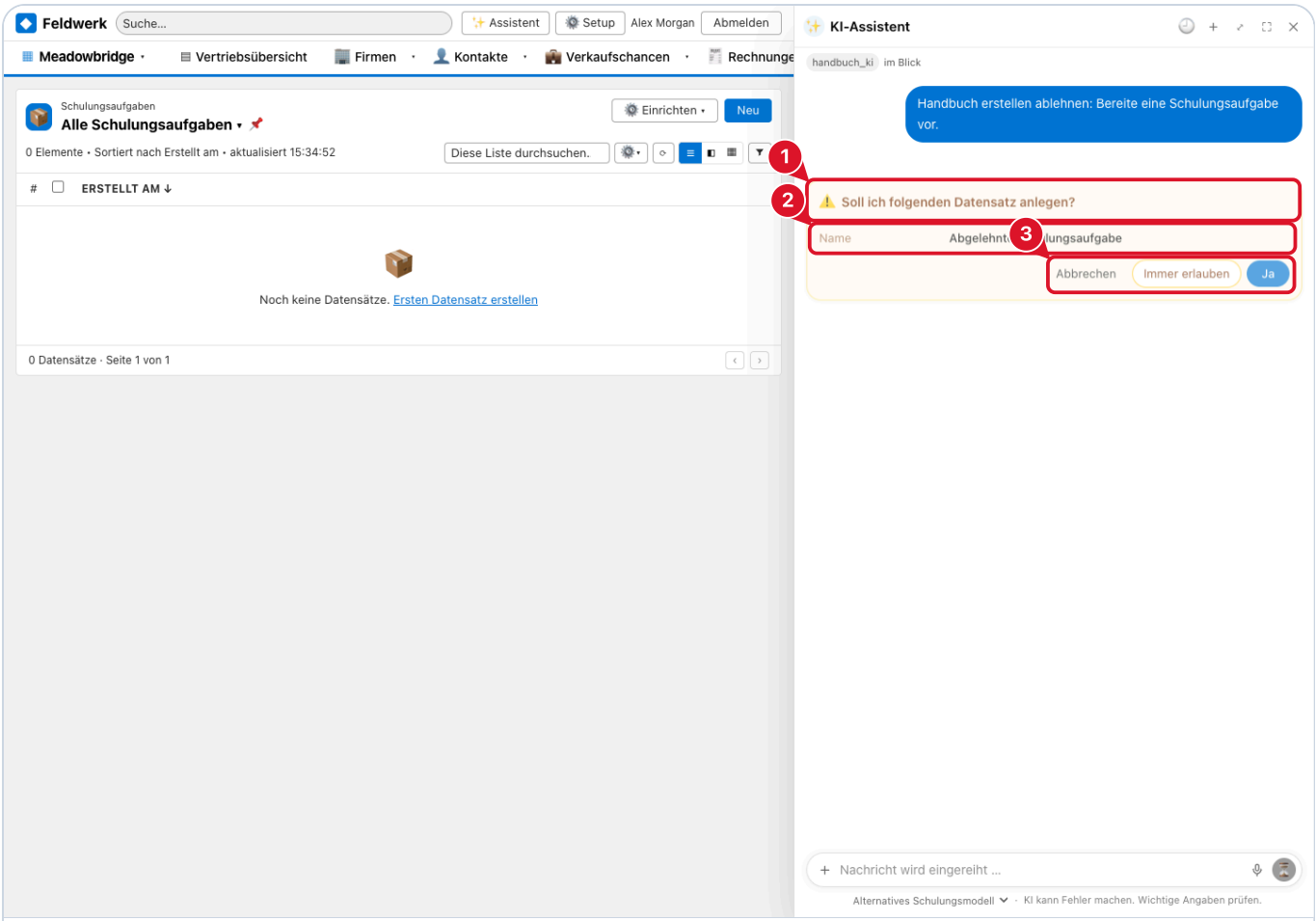
Beginnen Sie mit **Neuer Chat** ein separates Thema. Über den **Verlauf** können Sie frühere Gespräche öffnen, umbenennen oder löschen. Ein vergrößerter Bereich oder Vollbild bietet mehr Platz für längere Antworten.



1. Aktuellen Objektkontext prüfen.
2. Verlauf öffnen, neuen Chat starten oder Bereichsgröße ändern.
3. Anfrage eingeben, unterstützte Datei anhängen und senden.

Vorgeschlagene Aktionen prüfen

Benötigt ein Werkzeug eine Bestätigung, prüfen Sie die vorgeschlagene Operation und ihre Argumente. Genehmigen Sie nur, wenn Ziel und Werte Ihrer Anfrage entsprechen. Lehnen Sie falsche Vorschläge ab und beschreiben Sie die Korrektur. **Immer erlauben** erlaubt dieses Werkzeug für den Rest der Sitzung ohne erneute Nachfrage. Das ist weitergehend als die Genehmigung einer einzelnen Operation.



1. Vorgeschlagene Operation lesen.
2. Feldwert prüfen.
3. Abbrechen, dieses Werkzeug für die Sitzung erlauben oder diese eine Operation genehmigen.

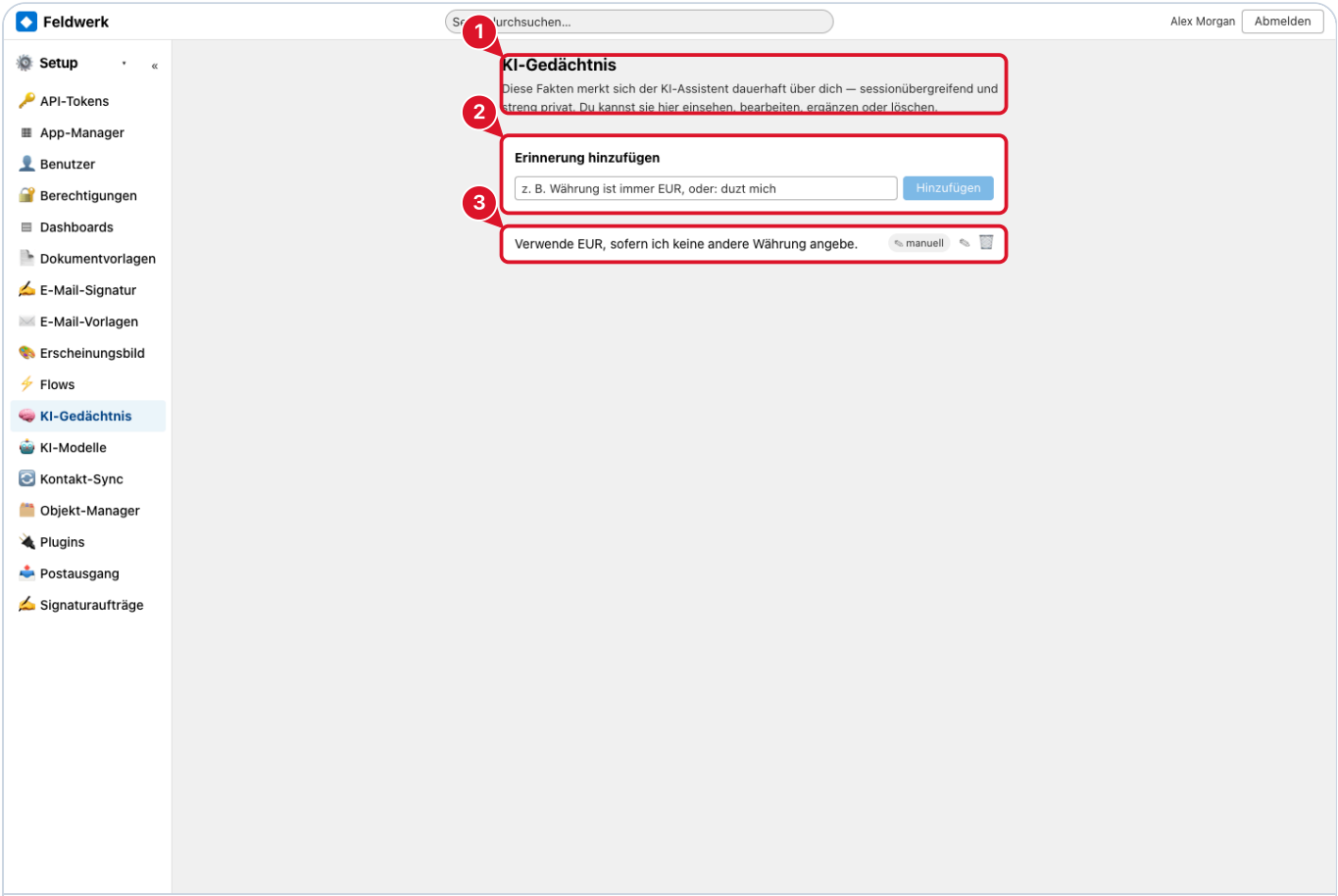
Mit **Stopp** beenden Sie eine laufende Antwort. Das macht bereits ausgeführte Änderungen nicht rückgängig. Prüfen Sie betroffene Datensätze, bevor Sie eine möglicherweise bereits abgeschlossene Anfrage wiederholen. Während der Bearbeitung eingegebene Nachrichten können in einer Warteschlange landen; entfernen Sie unerwünschte Nachrichten daraus, bevor sie gesendet werden.

Dateien und Diktat verwenden

Der Assistent unterstützt PNG-, JPEG- und WebP-Bilder, PDFs und Word-Dokumente (.docx) über die Anhangsauswahl oder durch Hineinziehen. Prüfen Sie die gewählten Dateien vor dem Senden; die aktuelle Oberfläche meldet nicht unterstützte Typen und Dateien über 10 MB. Jede Nachricht nimmt bis zu acht Dateien auf. PDF- und Word-Text wird auf dem Server extrahiert; eine gescannte PDF mit wenig auswählbarem Text muss gegebenenfalls als Bild bereitgestellt werden. Bilder benötigen ein konfiguriertes Sichtmodell. Der ausgewählte

Modellanbieter erhält den Gesprächskontext und die aufbereiteten Dateiinhalte. Diktat benötigt die Mikrofonberechtigung des Browsers und einen konfigurierten Transkriptionsdienst. Prüfen Sie die Transkription, bevor Sie sich auf Namen, Beträge oder Datumswerte verlassen.

Eine Datei am Chat ist etwas anderes als ein Anhang an einem Geschäftsdatensatz. Für Datensatzdateien siehe Anhänge und Themen. Prüfen Sie bei der Dokumenterkennung während der Datensatzerfassung die extrahierten Felder vor dem Speichern. Erkennung beweist nicht die Richtigkeit der Werte.



1. Erinnerungen gehören zu Ihrem Konto.
2. Eine dauerhafte Präferenz hinzufügen.
3. Gespeicherte Fakten prüfen und verwalten.

Gemerkte Präferenzen verwalten

1. Öffnen Sie **Setup → KI-Gedächtnis**.
2. Geben Sie eine dauerhafte Präferenz ein, etwa „Verwende EUR, wenn ich keine Währung angebe“, und wählen Sie **Hinzufügen**.

3. Korrigieren Sie einen gespeicherten Fakt über die Bearbeitungsaktion und speichern Sie.
4. Entfernen Sie einen Fakt, der den Assistenten nicht mehr leiten soll, über die Löschaktion und bestätigen Sie.

Erinnerungen bleiben über Gespräche hinweg erhalten und gehören zum aktuellen Benutzer. Das Löschen eines Gesprächs und einer Erinnerung sind separate Aktionen. Die Liste kennzeichnet, ob ein Fakt manuell oder durch den Assistenten ergänzt wurde. Halten Sie einzelne Kundenfakten an den jeweiligen Datensätzen, wo sie mit den übrigen Geschäftsdaten gepflegt werden können.

Dauerhaften fachlichen Kontext geben

Administratoren können den fachlichen KI-Systemprompt eines Objekts im Objekt-Manager pflegen. Erklären Sie, wofür das Objekt steht, wie Felder zu verstehen sind und welche Abläufe wichtig sind. Dieser Kontext begleitet relevante KI-/MCP-Werkzeugergebnisse. Schreiben Sie keine vorübergehenden Fakten eines einzelnen Datensatzes in einen objektweiten Prompt.

Der Dokumenterkennungs-Prompt hat einen anderen Zweck: Er beschreibt das Lesen typischer Quelldokumente und die Zuordnung ihrer Informationen zu Feldern. Halten Sie beide Prompts konkret und prüfen Sie Ergebnisse mit typischen Beispielen. Fachlicher Kontext setzt Anwendungsberechtigungen nicht außer Kraft.

Wenn eine Antwort oder Aktion fehlschlägt

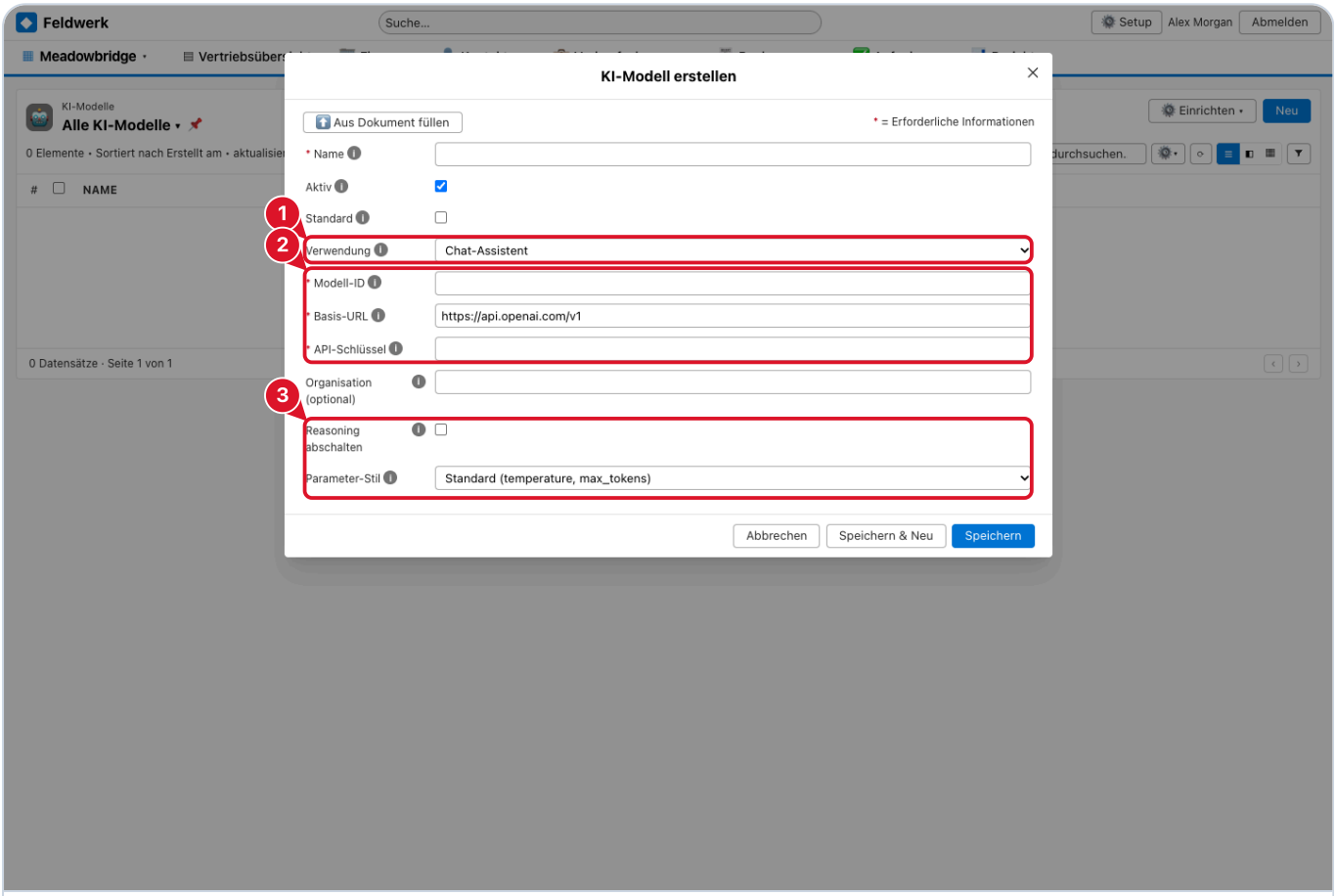
Prüfen Sie ausgewähltes Modell, Verbindung und Kontoberechtigungen. Wurde eine Antwort unterbrochen, prüfen Sie den aktuellen Datensatz, bevor Sie dieselbe Änderung erneut anfragen. Findet der Assistent ein Feld nicht, nennen Sie dessen genaue Beschriftung oder API-Namen und das gewünschte Objekt. Zu externen Assistenten und der Werkzeugsuche siehe API und MCP.

Ein Modell konfigurieren

1. Öffnen Sie **Setup → KI-Modelle** und wählen Sie **Neu**.
2. Geben Sie einen verständlichen Namen ein. Wählen Sie den Zweck: Chat für Gespräche und das Strukturieren extrahierter Texte oder Sicht für das Lesen von Bildern.
3. Geben Sie Modellkennung, kompatible Basis-URL und API-Schlüssel des Anbieters ein. Diese Werte müssen zum selben Anbieter gehören. Zeigen Sie niemals einen Schlüssel in Screenshots oder Supportberichten.

4. Wählen Sie den unterstützten Parameterstil und optionale Denkeinstellungen des Modells. Aktivieren Sie das Modell und wählen Sie es gegebenenfalls als Standard.
5. Speichern Sie und öffnen Sie den Assistenten erneut. Sind mehrere Chatmodelle verfügbar, wählen Sie eines über die Auswahl unter der Eingabe.
6. Testen Sie eine einfache rein lesende Frage und prüfen Sie anschließend eine vorgeschlagene Änderung, ohne sie zu genehmigen. Prüfen Sie sowohl die Anbieterantwort als auch die erlaubten Operationen Ihres Kontos.

Aktive vollständige Modelldatensätze haben Vorrang vor der älteren Umgebungskonfiguration des Servers. Ein nicht erreichbarer Anbieter bleibt ein Fehler, auch wenn sein Modell in der Auswahl erscheint. Die Administration muss die Verbindung korrigieren oder eine funktionierende Konfiguration wählen.

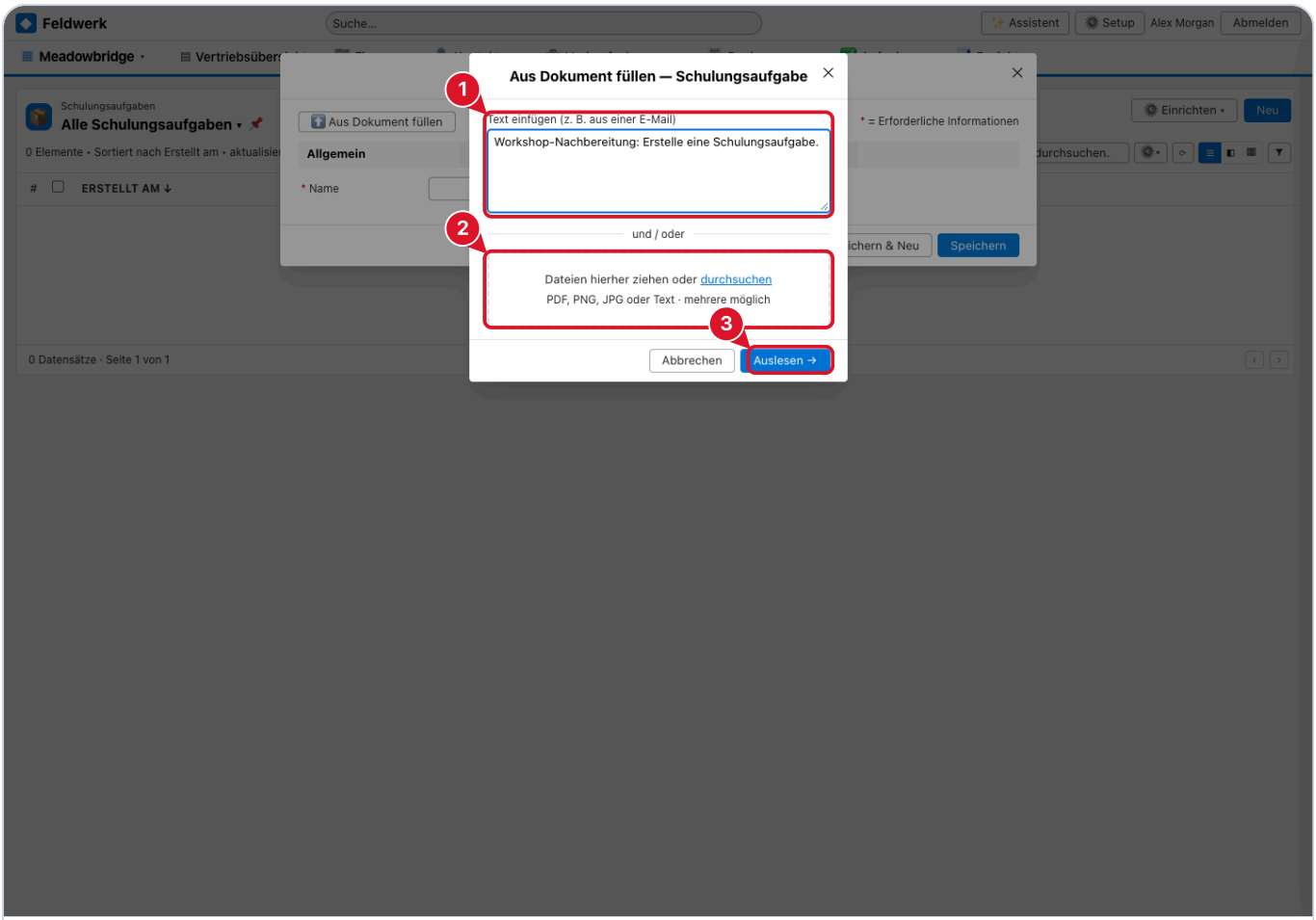


1. Verwendungszweck des Modells wählen.
2. Modell-ID, Basis-URL und API-Schlüssel des Anbieters konfigurieren.
3. Nur vom Modell unterstützte Optionen wählen.

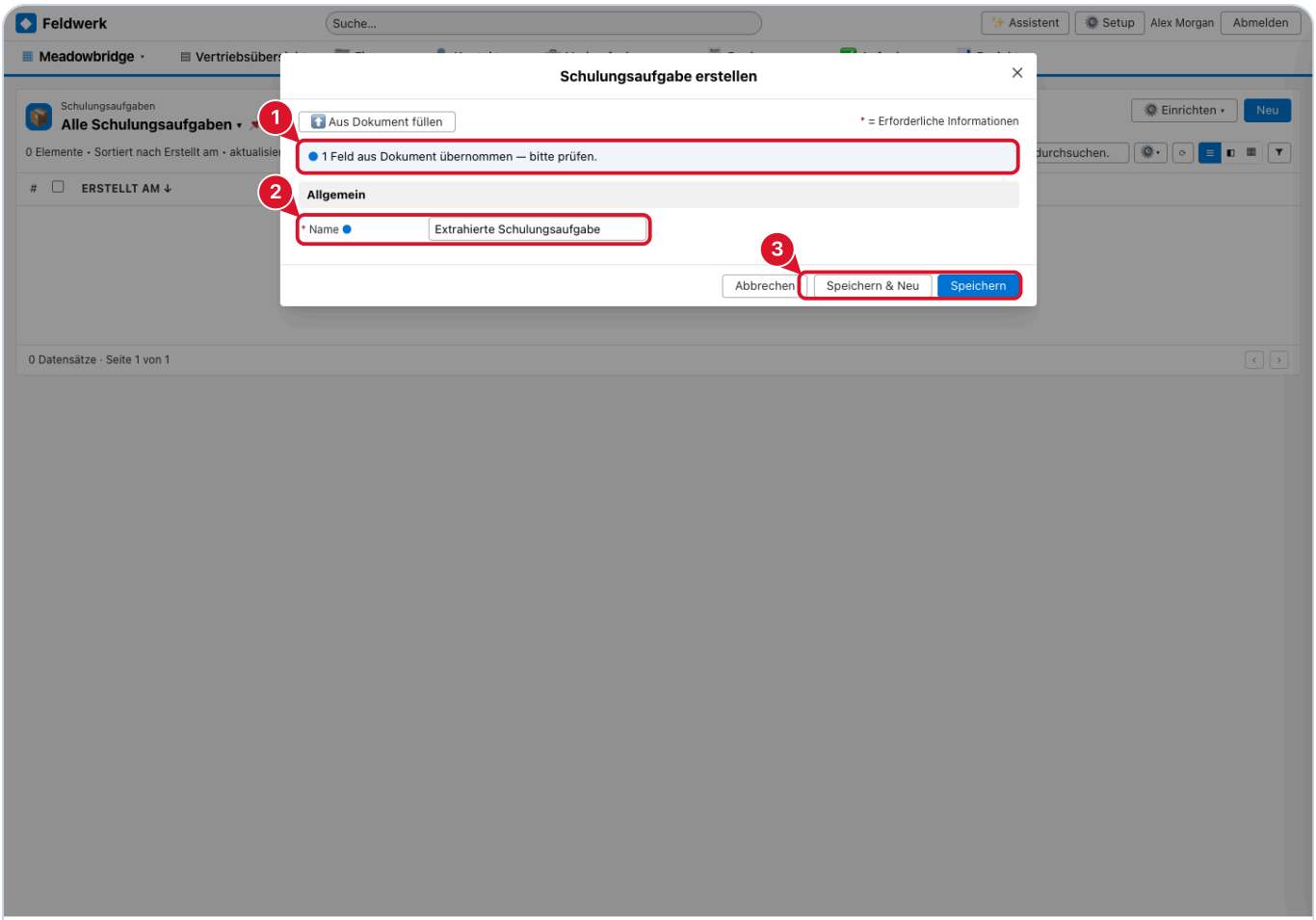
Extrahierte Werte vor dem Speichern prüfen

1. Öffnen Sie einen neuen Datensatz und wählen Sie **Aus Dokument füllen**.
2. Fügen Sie Quelltext ein oder wählen Sie die zu lesenden Dateien. Verwenden Sie die in diesem Dialog angegebenen Formate; seine Quellenoptionen unterscheiden sich von Chat-Anhängen.
3. Wählen Sie **Extrahieren** und warten Sie, bis das Formular ausgefüllt ist.
4. Prüfen Sie jedes erkannte Feld, Warnungen und vorgeschlagene Beziehungen. Ordnen Sie eine Beziehung dem gewünschten vorhandenen Datensatz zu, statt den vorgeschlagenen Namen als gespeicherten Link zu behandeln.
5. Korrigieren Sie die Felder und speichern Sie erst bei richtigem Ergebnis. Abbrechen lässt den ungespeicherten Datensatz unangelegt.

Die Extraktion schlägt nur Formularwerte vor. Sie speichert selbst keinen Datensatz. Objekt- und Feldberechtigungen gelten weiterhin. Der Dokumenterkennungs-Prompt im Objekt-Manager kann wiederkehrende Quellenformate erklären; er ist vom allgemeinen fachlichen Kontext des Objekts getrennt.



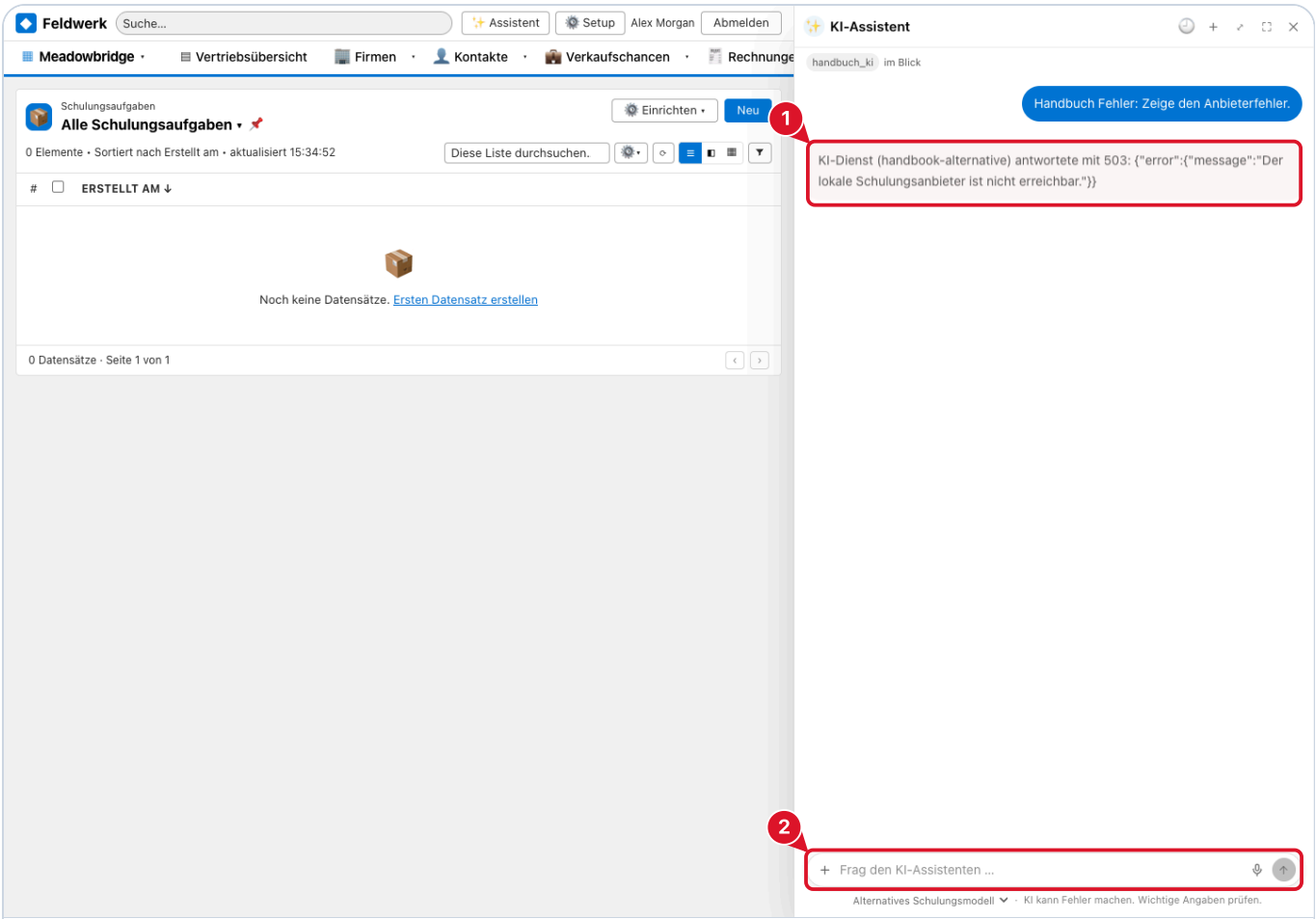
1. Quelltext einfügen.
2. Bei Bedarf unterstützte Dateien hinzufügen.
3. Feldvorschläge extrahieren.



1. Erkennungszusammenfassung lesen.
2. Markiertes Feld prüfen und gegebenenfalls korrigieren.
3. Datensatz speichern oder abbrechen; allein die Extraktion hat ihn nicht gespeichert.

Einen Anbieterfehler lesen

Im folgenden Beispiel liefert ein kontrollierter lokaler Anbieter absichtlich HTTP 503. Der Assistent zeigt den Fehler, damit Benutzer einen fehlgeschlagenen Anbieteraufruf von einer erfolgreichen Antwort unterscheiden können. Bitten Sie die Administration, Verfügbarkeit und Konfiguration zu prüfen. Kontrollieren Sie vor dem Wiederholen einer Änderung, ob ein früherer Werkzeugaufruf bereits abgeschlossen wurde.



1. Anbieterfehler und Status lesen.
2. Ergebnis vor einem erneuten Versuch prüfen.

Die Trainingsscreenshots verwenden einen lokalen deterministischen Anbieter. Sie zeigen Oberfläche und Bestätigungsablauf, messen jedoch nicht die Antwortqualität eines Modells.

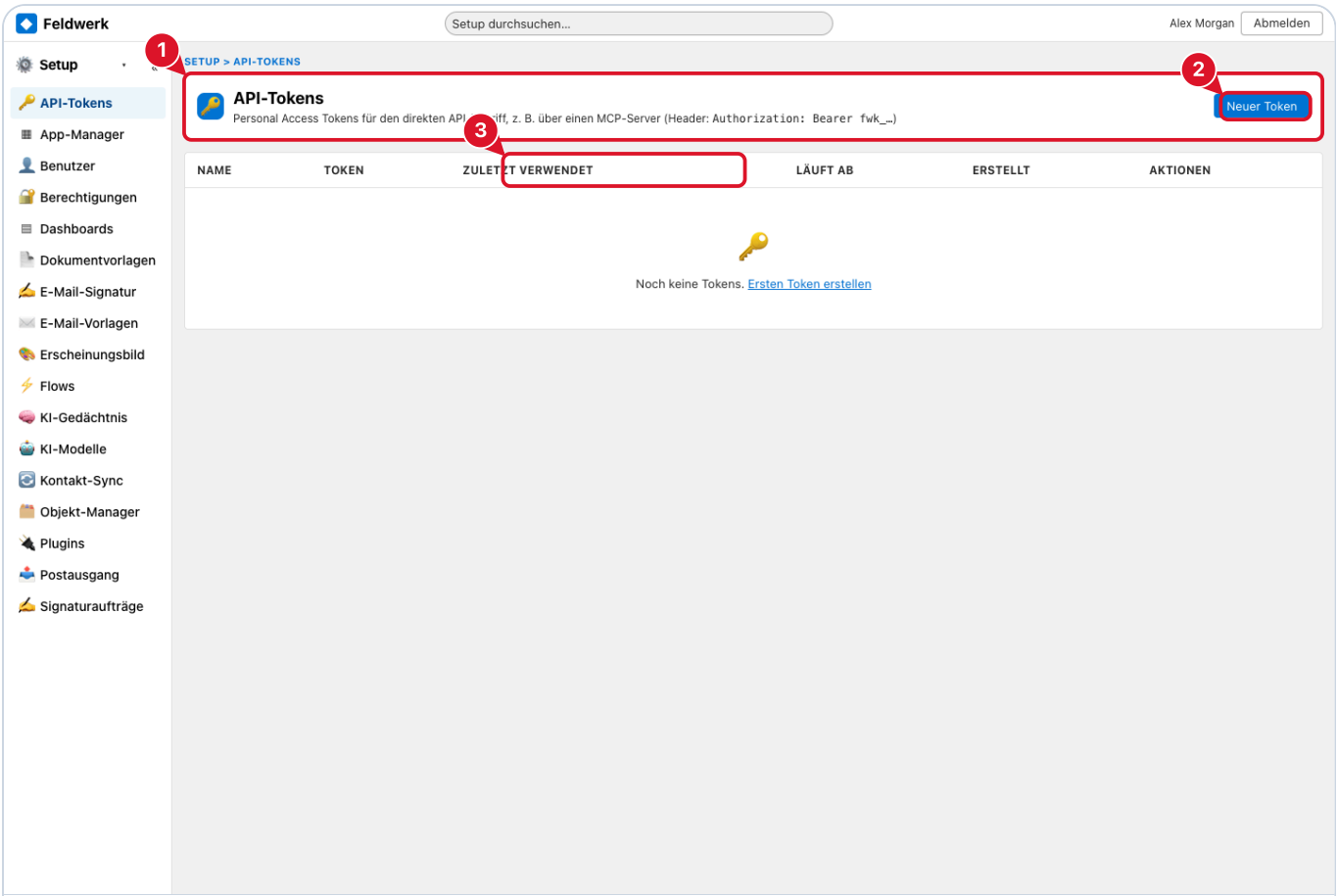
API, MCP und Integrationen

Assistenten anbinden, Datensätze importieren, Dateien hochladen und Kontaktsynchronisation konfigurieren.

Das Integrationskonto wählen

REST-API und Model Context Protocol (MCP) ermöglichen Zugriff auf Feldwerk ohne nachgebildete Browserklicks. Operationen laufen als das Konto, dem das Token gehört. Ein Mitgliedstoken wird nicht dadurch zum Administratortoken, dass ein Assistent es verwendet.

Verwenden Sie für eine dauerhafte Integration ein eigenes Konto mit den benötigten Objekt- und Feldberechtigungen. Erwägen Sie geschlossenen Feldzugriff, wenn nur ausdrücklich freigegebene Felder offengelegt werden sollen. Testen Sie mit diesem Konto vor der Anbindung. Siehe Benutzer, Berechtigungen und Freigaben.

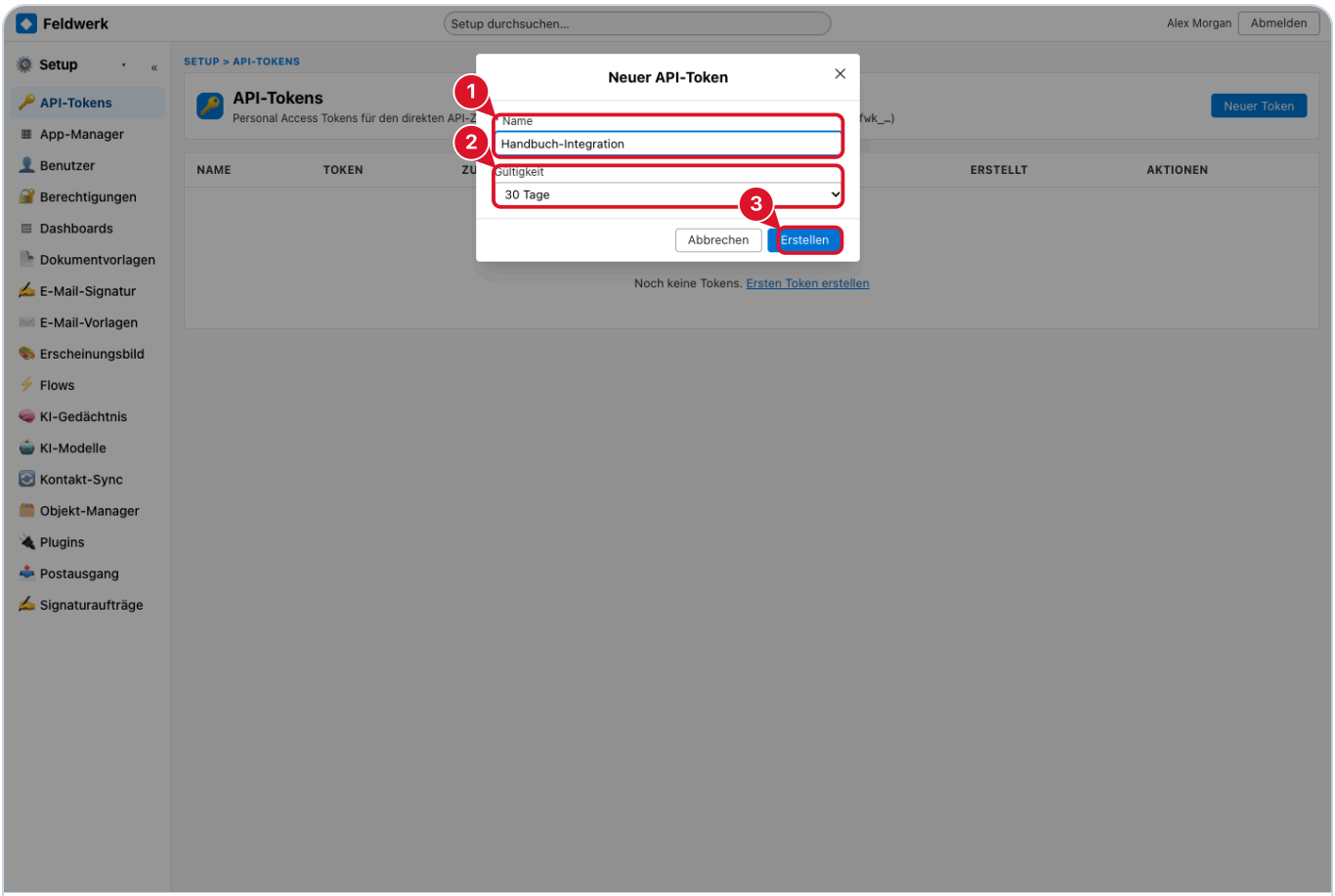


1. Tokens handeln als ihr Eigentümer.
2. Ein benanntes Token erstellen.
3. Letzte Nutzung und Ablauf prüfen.

Ein Token erstellen und widerrufen

1. Melden Sie sich mit dem Konto an, das die Integration verwenden soll.
2. Öffnen Sie **Setup → API-Tokens** und wählen Sie **Neues Token**.
3. Vergeben Sie einen zweckbezogenen Namen und wählen Sie eine passende Gültigkeitsdauer.
4. Kopieren Sie das angezeigte Token. Der vollständige Wert wird nur einmal zurückgegeben.
5. Speichern Sie es in den Zugangseinstellungen der Integration und verwenden Sie den Header `Authorization: Bearer YOUR_TOKEN`.
6. Prüfen Sie in der Tokenliste letzte Nutzung und Ablauf. Widerrufen Sie ein Token, wenn die Integration stillgelegt oder der Schlüssel ersetzt werden soll.

Jeder Benutzer verwaltet seine eigenen Tokens. Schreiben Sie kein echtes Token in Prompts, Screenshots, gemeinsam genutzte Skripte oder Quellcode-Repositories. Die folgenden Beispiele verwenden bewusst einen Platzhalter.



1. Integration benennen.
2. Gültigkeitsdauer festlegen.
3. Token erstellen; seinen geheimen Wert nur in Ihren Zugangsspeicher kopieren.

Über die REST-API lesen

Verwenden Sie die Feldwerk-Adresse Ihrer Organisation als Basis-URL. Mit `GET /api/objekte` und dem Autorisierungsheader finden Sie beispielsweise zugängliche Objekte. Lesen Sie die Felddefinitionen des Objekts, bevor Sie Datensatzwerte zusammenstellen. Objektbeschriftungen helfen Menschen bei der Navigation; Feld-API-Namen sind die Schlüssel im `werte`-Objekt eines Datensatzes.

Datensatzschreibvorgänge verwenden die reguläre Validierungs- und Automatisierungsverarbeitung. Ein Feld kann Pflicht, berechnet, eingeschränkt oder einer Validierungsregel unterworfen sein. Behandeln Sie zurückgegebene Fehler, statt aus einer

erfolgreichen Verbindung auf die Erlaubnis jeder Operation zu schließen. Verwenden Sie beim Lesen von Listen die Seitennavigation.

Einen MCP-Client verbinden

1. Wählen Sie einen Client mit Unterstützung für einen HTTP-MCP-Server mit Bearer-Token-Authentifizierung.
2. Setzen Sie die Server-URL auf `https://YOUR-FELDWERK-HOST/mcp`.
3. Hinterlegen Sie das Bearer-Token im Zugangsdatenfeld des Clients.
4. Verbinden Sie den Client und lassen Sie ihn initialisieren und Werkzeuge ermitteln.
5. Beginnen Sie mit einer rein lesenden Aufgabe, etwa dem Auflisten zugänglicher Objekte.

Der Server verwendet HTTP POST und ist zustandslos. Das Öffnen von `/mcp` als normale Browserseite ist kein Verbindungstest. Werkzeugnamen und API-Kennungen bleiben technische deutsche Namen, auch bei englischer Oberfläche; verwenden Sie die ermittelten Namen exakt.

Vor dem Schreiben erkunden

1. Rufen Sie `objekte_auflisten` auf, um den Objekt-API-Namen zu ermitteln.
2. Rufen Sie für dieses Objekt `felder_auflisten` auf und prüfen Sie Feld-API-Namen und Typen.
3. Rufen Sie `datensaetze_auflisten` mit geeignetem Filter oder passender Seitengröße auf.
4. Lesen Sie vor einer vorgeschlagenen Änderung den vorgesehenen Datensatz. Verwenden Sie eine konkrete Datensatz-ID, statt einen eindeutigen Namen vorauszusetzen.
5. Lesen Sie nach einem erlaubten Schreibvorgang das Ergebnis erneut, einschließlich berechneter Werte und relevanter Automatisierungswirkungen.

Für das Beispielobjekt Verkaufschancen lautet ein Werkzeugaufruf `{"name":"datensaetze_auflisten","arguments":{"objekt":"chance"}}`. Der lokale Handbuchclient prüft, dass ein Mitglied mit Lesezugriff diese Datensätze ohne das eingeschränkte Feld `wert` erhält. Ein API-Bearbeitungsversuch desselben Kontos wird abgelehnt.

Generische API-Werkzeuge verwenden

Deckt kein spezielles Werkzeug eine Operation ab, finden Sie mit `api_operationen_auflisten` eine registrierte Route und prüfen mit `api_operation_lesen` ihr Schema. Fehlende ältere Schemadokumentation wird ausdrücklich gekennzeichnet; erraten Sie keinen unbekannten Anfragekörper. Übergeben Sie genaue Routenvorlage, Parameter und Anfragekörper an `api_aufrufen`.

Große Ergebnisse werden abschnittsweise zurückgegeben. Lesen Sie mit `api_ergebnis_lesen` und dem gelieferten nächsten Offset die gespeicherte Antwort weiter. Das führt die ursprüngliche Operation nicht erneut aus. Der Zwischenspeicher gilt höchstens zehn Minuten und kann früher geleert werden; er ist ein Schnappschuss der ursprünglich autorisierten Antwort, keine neue Abfrage nach Berechtigungsänderungen. Wiederholen Sie keine Mutation automatisch, nur weil ihre gespeicherte Antwort abgelaufen ist.

Binäre Antworten liefern zunächst Metadaten und einen Downloadlink. Fordern Sie Binärinhalte nur bei Bedarf ausdrücklich an. Die generische Brücke führt keine OAuth-Einwilligung im Browser durch, ändert keine Authentifizierungssitzung und legt keine neu erzeugten geheimen Tokens im Modellkontext offen. Konfigurieren Sie Zugangsdaten über den vorgesehenen Benutzerkanal.

Verbindungs- und Berechtigungsfehler untersuchen

Prüfen Sie bei Authentifizierungsfehlern Basis-URL, Bearer-Header, Tokenablauf und Widerruf. Prüfen Sie bei verbotenen Operationen Berechtigungen des Tokeninhabers und Datensatzfreigaben. Ein Werkzeugergebnis kann einen Operationsfehler melden, obwohl die HTTP-Anfrage erfolgreich war; prüfen Sie `isError` und Meldung im MCP-Ergebnis. Lesen Sie nach einem Schreib-Timeout den Zieldatensatz vor einer Wiederholung, um doppelte Arbeit zu vermeiden.

Eine Datei über eine Integration hochladen

Dateien gehören zu Datensätzen und erfordern Bearbeitungsrecht am Ziel. Für kleine Dateien akzeptiert das MCP-Werkzeug `anhang_hochladen` Base64-Inhalt. Verwenden Sie bei größeren Dateien `anhang_upload_ticket`, damit der Client direkt übertragen kann, ohne den Dateiinhalt in einen Assistentenkontext zu stellen.

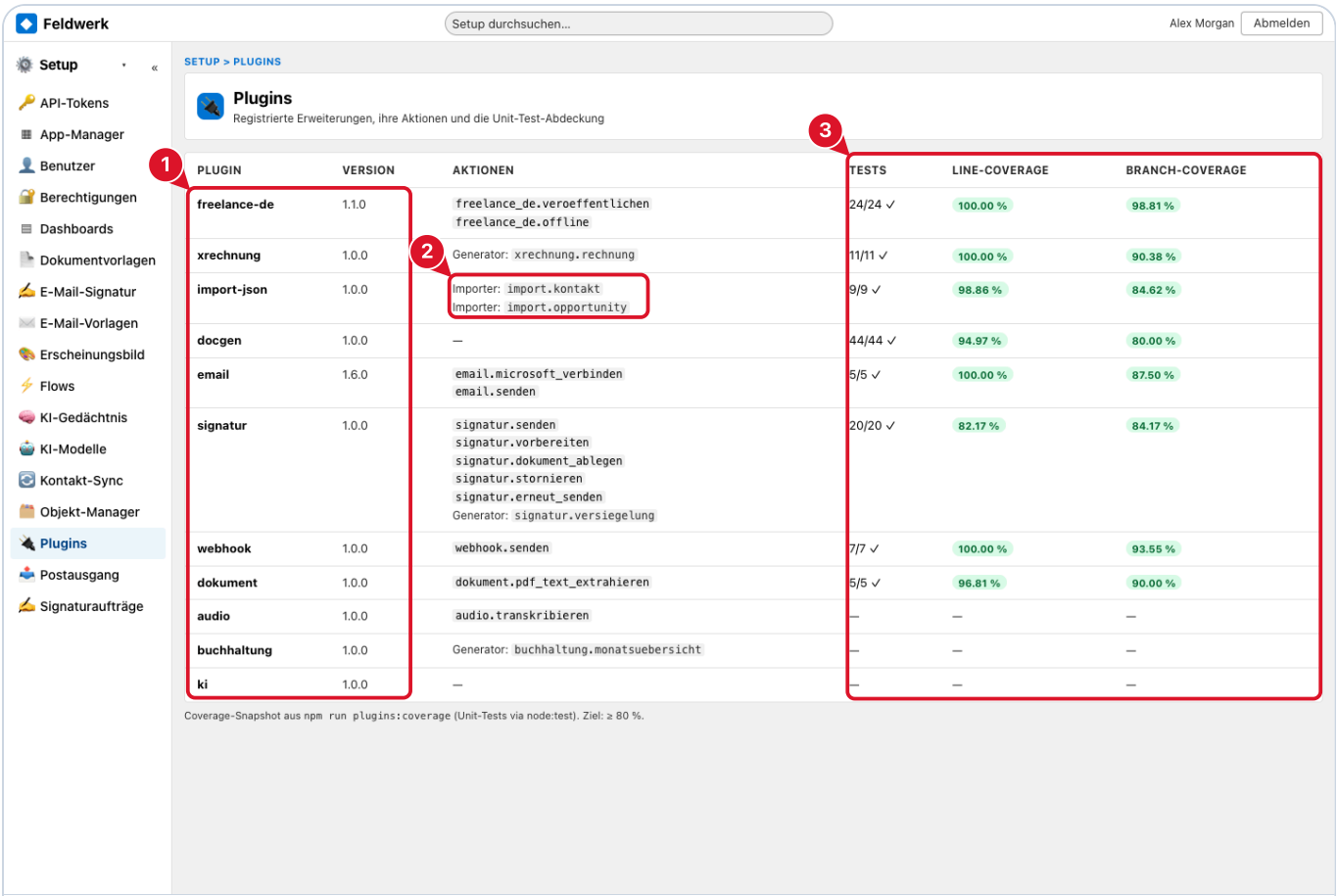
1. Ermitteln Sie den Zieldatensatz und fordern Sie ein Uploadticket an. Die entsprechende REST-Anfrage lautet `POST /api/anhaenge/upload-ticket` mit `{"datensatzId":"RECORD_ID"}`.
2. Verwenden Sie die zurückgegebene `uploadUrl`. Senden Sie einen Multipart-POST mit der Datei im durch `feld` bezeichneten Feld (derzeit `datei`) und dem Ticket im durch `header` bezeichneten Header (derzeit `X-Upload-Ticket`).
3. Prüfen Sie die Erfolgsantwort und lesen Sie die Anhangliste des Datensatzes. Laden Sie den Anhang bei benötigter Inhaltsprüfung über seine normale autorisierte Downloadroute herunter.

Ein Ticket ist ein Zugangsschlüssel: Halten Sie es vertraulich. Es läuft nach 15 Minuten ab, geht beim Serverneustart verloren und kann nur einmal verwendet werden. Auch ein fehlgeschlagener Uploadversuch verbraucht es; fordern Sie vor einer Wiederholung ein neues Ticket an. Die Grenze für Datensatzanhänge beträgt 25 MiB. Der Handbuchtest lädt eine kleine Textdatei hoch, vergleicht die heruntergeladenen Bytes und prüft die Ablehnung einer erneuten Ticketverwendung.

Registrierte Plugins prüfen

Öffnen Sie als Administrator **Setup → Plugins**. Der Katalog zeigt registrierte Erweiterungen dieses Servers mit Versionen, Aktionskennungen, Generatoren, Importern und Feeds. Er ist ein Verzeichnis, keine Installations- oder Aktivierungsoberfläche. Konfiguration gehört in die eigenen Einstellungen oder Datensätze der Erweiterung. Beispielsweise bedeutet ein E-Mail-Plugin-Eintrag nicht, dass ein Postfach verbunden ist.

Verwenden Sie bei der Aktionssuche über die API oder der Automatisierungskonfiguration die exakten technischen Kennungen aus dem Katalog. Eine Testabdeckungsquote ist ein Stand automatisierter Codetests, keine Zustandsprüfung des externen Live-Diensts. Ein Strich belegt keine defekte Verbindung.



1. Installierte Erweiterung und Version erkennen.
2. Exakte Importerkennung kopieren.
3. Testabdeckung als Codetest-Stand verstehen, nicht als Live-Verbindungsstatus.

JSON-Datensätze importieren

Das eingebaute Plugin `import-json` bietet benannte Importer, keinen allgemeinen CSV-Uploadassistenten. Administratoren können ein JSON-Objekt oder ein Array an `POST /api/import/IMPORTER_NAME` senden. Registrierte Namen finden Sie über `GET /api/plugins`. Importschreibvorgänge folgen den normalen Berechtigungen, Feldvalidierungen und Automatisierungen.

`import.opportunity` zielt auf das Objekt mit dem exakten API-Namen `opportunity`. Es verwendet nicht automatisch das Handbuchobjekt `chance`, nur weil dessen Beschriftung Verkaufschancen lautet. Bereiten Sie vor dem Import das Zielschema vor. Die Zuordnung umfasst `Opportunity_Name` → `name` und `External_ID` → `externe_id`. Eine externe ID wird als Aktualisierungsschlüssel bevorzugt; `SF_ID`/`sf_id` werden als Ersatzschlüssel unterstützt. Technische Quellschlüssel unterscheiden Groß- und Kleinschreibung und werden nicht mit der Oberfläche übersetzt.

```
POST /api/import/import.opportunity
[
  {
    "Opportunity_Name": "Service planning",
    "External_ID": "example-001"
  }
]
```

Existiert ein passender Schlüssel, aktualisiert der Importer den Datensatz; andernfalls erstellt er einen. Verwenden Sie einen stabilen eindeutigen Schlüssel und konfigurieren Sie bei Bedarf Eindeutigkeit. Ein fehlender Schlüssel ist ein Fehler. Leere Quellwerte werden übersprungen; mit diesem Importer lassen sich gespeicherte Felder daher nicht leeren.

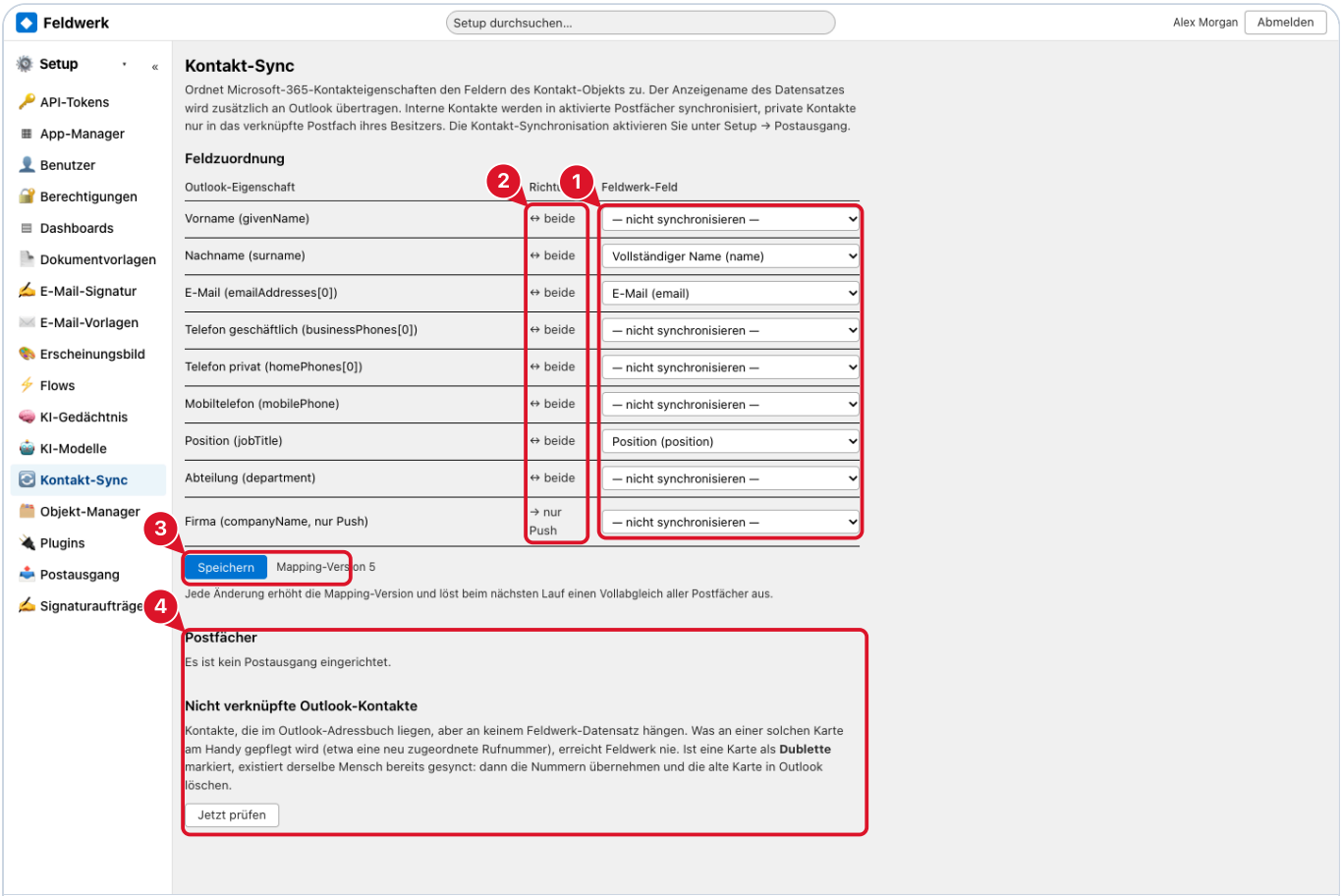
`import.kontakt` zielt auf `kontakt`. Es verwendet eine mitgelieferte externe ID, sonst die E-Mail-Adresse. Seine vordefinierte Quellzuordnung umfasst `Nachname`, `Vorname` und `E-Mail-Adresse` und ergänzt Standardwerte für Kontakttyp und Status. Das Ziel muss die erwarteten Felder enthalten; Firmenbeziehungen werden nicht automatisch aufgelöst. Prüfen Sie die Zuordnung des installierten Importers vor dem Einsatz mit Ihrem Datenmodell.

1. Bereiten Sie Zielobjekt, Felder und Aktualisierungsschlüssel vor. Beginnen Sie mit einer kleinen typischen Datenmenge.
2. Senden Sie das JSON an den gewählten Importer.
3. Prüfen Sie `erstellt`, `aktualisiert` und jeden Eintrag in `fehler`.
4. Finden Sie die Eingabezeile über `zeile` im Fehler; die Zählung beginnt bei 1. Ein Fehler der Gesamteingabe kann Zeile 0 verwenden.
5. Lesen Sie erfolgreich importierte Datensätze einschließlich berechneter Felder und Automatisierungswirkungen erneut. Korrigieren Sie fehlgeschlagene Zeilen vor einer Wiederholung.

Eine HTTP-200-Antwort kann Teilerfolge enthalten. Erfolgreiche Zeilen bleiben gespeichert, wenn eine andere scheitert. Behandeln Sie die Antwort nicht als Alles-oder-nichts-Transaktion und senden Sie keine Datenmenge mit instabilen Schlüsseln ungeprüft erneut. Der Zweizeilentest des Handbuchs erstellt einen Datensatz, meldet die andere Zeile als fehlerhaft und aktualisiert beim zweiten Import den bestehenden Datensatz ohne Duplikat.

Microsoft-365-Kontaktsynchronisation konfigurieren

Die Kontaktsynchronisation verwendet das Objekt mit API-Namen `kontakt`. Öffnen Sie als Administrator **Setup → Kontakt-Sync**, um Felder zuzuordnen und den Postfachstatus zu prüfen. Der Screenshot zeigt die lokale Konfiguration ohne verbundenes externes Postfach und belegt keine laufende Microsoft-365-Synchronisation.



1. Feld wählen oder Eigenschaft deaktivieren.
2. Beidseitige und nur ausgehende Synchronisation unterscheiden.
3. Speichern und Zuordnungsversion prüfen.
4. Postfachstatus und unverknüpfte Outlook-Kontakte prüfen.

1. Prüfen Sie Kontaktobjekt und exportierbare Informationen. Der Synchronisationsdienst verwendet eine Systemidentität; die Feldbeschränkungen eines Postfachbenutzers sind kein Exportfilter. Die Zuordnung bestimmt ausdrücklich die zu synchronisierenden Kontakteigenschaften.
2. Wählen Sie je Outlook-Eigenschaft ein kompatibles Feldwerk-Feld oder **— nicht synchronisieren —**. Verfügbar sind Vor- und Nachname, eine E-Mail-Adresse,

geschäftliche und private Telefonnummer, Mobiltelefon, Position, Abteilung und Firma.

3. Lesen Sie die Richtungsspalte. Die meisten Eigenschaften synchronisieren in beide Richtungen. Firma wird nur von Feldwerk nach Outlook übertragen; ein Nachschlagefeld kann den Anzeigenamen des verknüpften Datensatzes liefern. Außerhalb der konfigurierbaren Eigenschaften wird auch der eigene Kontaktanzeigename als Outlook-Anzeigename und Ablagename gesendet.
4. Wählen Sie **Speichern**. Eine geänderte Zuordnung erhöht ihre Version und plant einen vollständigen Abgleich beim nächsten Worker-Lauf. Unverändertes Speichern erhöht die Version nicht.
5. Verwenden Sie unter **Setup → Postausgang** ein aktives, mit Microsoft verbundenes Postfach und aktivieren Sie **Kontakte synchronisieren**. Die Microsoft-Verbindung benötigt Kontaktzugriff (`Contacts.ReadWrite`). Verlangt der Status eine erneute Autorisierung, verbinden Sie das Postfach erneut über den Microsoft-Einwilligungsablauf.
6. Kehren Sie zu Kontakt-Sync zurück und prüfen Sie letzten Lauf, Zuordnungsversion, Anzahl aktiver/getrennter/fehlgeschlagener Verknüpfungen und letzten Fehler. Eine ausstehende Version bedeutet, dass das Postfach die neueste Zuordnung noch nicht abgeglichen hat.

In der Konfigurations-API ersetzt `PUT /api/kontakt-sync/konfig` die vollständige Zuordnung. Ausgelassene Eigenschaften werden deaktiviert; senden Sie die gesamte gewünschte Zuordnung statt einer Einzeländerung. Der Server lehnt inkompatible Feldtypen ab. Ein gespeichertes Feld, das fehlt oder inzwischen einen inkompatiblen Typ hat, wird in der Auswahl entsprechend gekennzeichnet; wählen Sie einen gültigen Ersatz.

Welche Kontakte gelangen wohin?

Ein interner Kontakt, auch mit leerer Sichtbarkeit, kommt für jedes Postfach mit aktivem Kontakt-Sync infrage. Ein privater Kontakt kommt nur für das verknüpfte Postfach seines Eigentümers infrage. Dieses Kontaktsichtbarkeitsfeld ändert nicht selbst, wer den Datensatz innerhalb von Feldwerk lesen darf. Konfigurieren Sie Objektstandards und Freigaben separat; siehe Benutzer, Berechtigungen und Freigaben.

Erstellen Sie neue Kontakte in Feldwerk. Die Synchronisation erstellt ihre Outlook-Gegenstücke und tauscht Änderungen verknüpfter Kontakte aus. Eine ausschließlich in Outlook neu angelegte Karte wird nicht automatisch nach Feldwerk importiert. Beim ersten Abgleich kann eine vorhandene Outlook-Karte übernommen werden, wenn ihre E-Mail-Adresse oder danach ihr Name einen eindeutigen Treffer ergibt. Mehrdeutige Treffer werden nicht geraten.

Änderungen, Konflikte und Löschen

Änderungen verschiedener Felder können zusammengeführt werden. Ändern beide Seiten dasselbe synchronisierte Feld, gewinnt Feldwerk. Firma und Datensatzanzeigename bleiben durch Feldwerk bestimmt. Das Löschen einer verknüpften Outlook-Karte trennt die Verknüpfung; es löscht weder den Feldwerk-Datensatz noch erstellt es die Karte automatisch neu. Auch das Verschieben aus dem unterstützten Kontaktordner kann die Verknüpfung trennen. Bitten Sie die Administration, den Verknüpfungsstatus zu untersuchen.

Das Löschen eines Kontakts in Feldwerk entfernt seine synchronisierten Gegenstücke. Der Wechsel von intern zu privat entfernt Kopien aus anderen zuvor berechtigten Postfächern. Das Abschalten von **Kontakte synchronisieren** pausiert die Verarbeitung, entfernt aber keine bereits vorhandenen Outlook-Kopien. Freigegebene Postfächer, Kontaktunterordner, Fotos, Notizen, Adressen und Geburtstage gehören nicht zum unterstützten Umfang dieser Synchronisation.

Eine unverknüpfte Outlook-Karte untersuchen

Wählen Sie unter **Unverknüpfte Outlook-Kontakte** die Aktion **Jetzt prüfen**, um aktivierte Postfächer bei Bedarf auszulesen. Diese Diagnose ist kein Stapelimporter. Ein Duplikathinweis verlinkt den entsprechenden Feldwerk-Datensatz. Vergleichen Sie Informationen und sichern Sie benötigte Details, bevor Sie eine veraltete Outlook-Karte entfernen. Änderungen an einer unverknüpften Karte gelangen nicht nach Feldwerk. Hat kein Postfach Kontakt-Sync aktiviert, sagt die Oberfläche dies ausdrücklich, statt eine erfolgreiche externe Synchronisation zu melden.

Fehlerbehebung

Häufige Probleme anhand von Meldungen, Konfiguration und tatsächlich gespeichertem Ergebnis untersuchen.

Mit dem genauen Ergebnis beginnen

Lesen Sie die Meldung vor einem erneuten Versuch. Eine hilfreiche Problembeschreibung nennt Schritte, erwartetes Ergebnis, tatsächliche Meldung, Objekt- und Datensatzreferenz, Zeitpunkt und Oberflächensprache. Geben Sie an, ob ein oder mehrere Datensätze betroffen sind und ob etwas gespeichert, heruntergeladen oder eingereiht wurde. Lassen Sie Zugangsdaten und private Nachrichteninhalte weg.

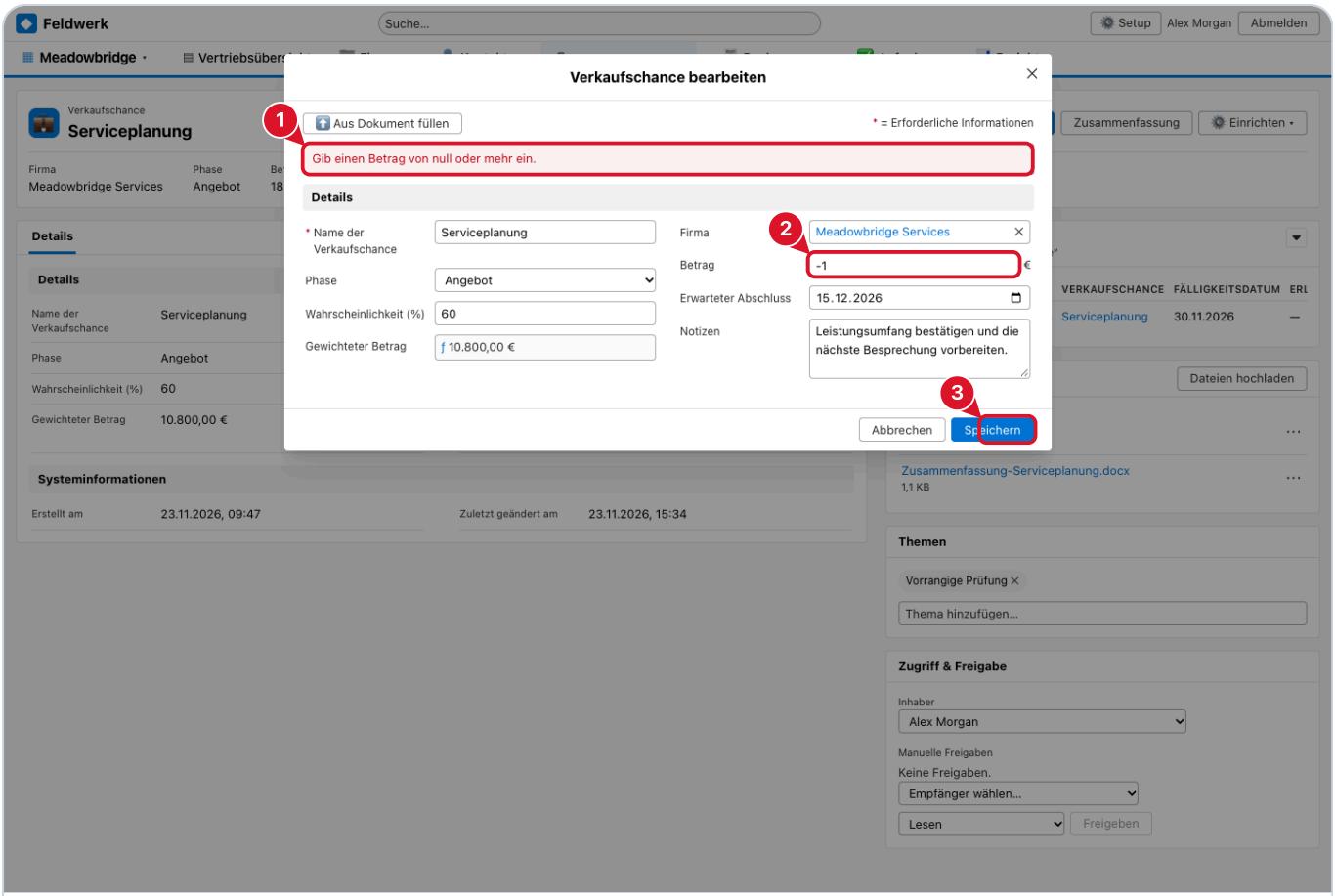
Stellen Sie das Problem mit einem harmlosen Testdatensatz nach. Öffnen Sie den gespeicherten Datensatz erneut, prüfen Sie den Anhang oder den Postausgang, um das Ergebnis zu bestätigen. Die Administration kann anschließend passende Konfigurationen und Protokolle prüfen.

Anmelde- oder Zugriffsprobleme

Prüfen Sie, ob Sie die Feldwerk-Adresse Ihrer Organisation und das richtige Konto verwenden. Werden Zugangsdaten abgelehnt, bitten Sie die Administration, das Konto zu prüfen. Fehlt nach der Anmeldung ein Register oder Objekt, prüfen Sie gewählte App und Kontoberechtigungen. Ein fehlender Navigationseintrag beweist keine Datenlöschung.

Ein Datensatz lässt sich nicht speichern

1. Prüfen Sie Pflichtfelder und Eingabeformate.
2. Lesen Sie die konkrete Validierungsmeldung und korrigieren Sie den angegebenen Wert.
3. Prüfen Sie, ob ein eindeutiger Wert bereits in einem anderen Datensatz verwendet wird.
4. Weist die Meldung auf eine gleichzeitige Änderung hin, laden Sie die aktuelle Version und gleichen Sie Ihre Änderungen damit ab.
5. Bleibt der Fehler bestehen, bitten Sie die Administration, Regeln und ausgelöste Automatisierungen zu prüfen.



1. Konkreten Korrekturhinweis lesen.
2. Eingabe korrigieren.
3. Speichern und Ergebnis prüfen.

Im Beispiel wird Betrag -1 mit „Geben Sie einen Betrag von null oder mehr ein.“ abgelehnt. Nach Korrektur auf 18000 ist das Speichern möglich. Der versuchte ungültige Wert ist nicht der gespeicherte Wert.

Fehlende Datensätze oder Felder

Leeren Sie die Listensuche und prüfen Sie Filter, Eigentümerbereich und Beziehungsumfang. Versuchen Sie die globale Suche mit mindestens zwei Zeichen. Prüfen Sie Feldzugriff und Datensatzsichtbarkeit. Ein Datei-Upload macht Dokumentinhalte nicht automatisch durchsuchbar; die Extraktion in ein durchsuchbares Datensatzfeld muss konfiguriert sein.

Unerwartete Berechnungen

Prüfen Sie bei einer Formel Feld-API-Namen, Rückgabetyp und Leerwertbehandlung. Prüfen Sie bei Aggregaten das untergeordnete Nachschlagefeld, das ausgewählte Wertfeld und die Filter. Unterscheiden Sie bei Berichten eine begrenzte Vorschau von einer vollständigen

Ausführung und prüfen Sie Gruppierung und Aggregation. Überschreiben Sie kein berechnetes Feld manuell, um falsche Eingangswerte zu verdecken.

Dateien und Dokumente

Prüfen Sie bei Uploadfehlern die Größenbegrenzung pro Datei und das Ergebnis jeder gewählten Datei. Einige Dateien können erfolgreich hochgeladen worden sein, obwohl eine andere fehlgeschlagen ist. Prüfen Sie Zieldatensatz und Bereichsquelle.

Prüfen Sie bei erzeugten Dokumenten Quellobjekt, Platzhalter und Voraussetzung der Vorlage. Die PDF-Umwandlung benötigt außerdem den konfigurierten Serverkonverter. Ein erfolgreicher Download und die erfolgreiche Speicherung als Anhang sind getrennte Ergebnisse; lesen Sie die Ergebnismeldung.

Eine Automatisierung lief nicht oder änderte nichts

Prüfen Sie aktive Version, Auslöseereignis und Eintrittsbedingungen. Flow-Bedingungen verwenden den aktuellen Datensatzkontext, etwa `$Datensatz.feld`. Prüfen Sie die Ausführung dieser Version und vergleichen Sie ihren Pfad mit dem erwarteten Zweig.

Ändert ein Schritt Datensätze aktualisieren nichts, prüfen Sie Zieldatensatzressource und Feldzuordnung. Prüfen Sie bei geplanten Flows Schedulerbetrieb, Zeitzone, Zeitplan und Datensatzfilter. Testen Sie Bildschirm-Flows mit den Berechtigungen der tatsächlichen Benutzer. Fehler nach dem Speichern machen den ursprünglich gespeicherten Datensatz nicht zwingend rückgängig.

Eine E-Mail ist nicht angekommen

Unterscheiden Sie zunächst Entwurf, eingereiht, verzögert, fehlgeschlagen und gesendet in den verfügbaren E-Mail- und Postausgangsansichten. Prüfen Sie Empfänger, konfigurierte Absenderverbindung, vorgesehenes Versandfenster und letzten Fehler. Eine eingereihte Nachricht beweist keine Zustellung. Reihen Sie bei der Untersuchung eines Transportproblems nicht wiederholt dieselbe Nachricht ein.

KI ist nicht verfügbar oder liefert unerwartete Ergebnisse

Prüfen Sie, ob ein aktives Modell für die benötigte Fähigkeit konfiguriert ist und seine Anbieteranbindung funktioniert. Prüfen Sie ausgewähltes Modell, bereitgestellten Kontext und Werkzeugergebnis. Vergleichen Sie erzeugte Antworten mit dem zugrunde liegenden Datensatz oder Dokument. Fügen Sie keine Anbieterschlüssel in Supportnachrichten oder Screenshots ein.

Probleme bei der Kontaktsynchronisation

Prüfen Sie konfigurierte Richtung, Objektzuordnung, Verbindung und letzten Synchronisationsstatus. Klären Sie vor einem erneuten Schreibversuch, welche Seite für den jeweiligen Wert maßgeblich ist. Scheitert ein externer Dienst, bewahren Sie die Fehlermeldung auf und benennen Sie ein einzelnes betroffenes Beispiel, statt pauschale Änderungen an allen Kontakten vorzunehmen.

Formelreferenz

Argumente und geprüfte Beispiele für alle 94 verfügbaren Funktionen. Verwenden Sie den verlinkten Katalog unter Formelfelder zum Finden einer Funktion.

Die Beispiele lesen

Eckige Klammern kennzeichnen optionale Argumente; geben Sie diese Klammern nicht mit ein. Trennen Sie Argumente durch Kommas, setzen Sie Text in Anführungszeichen und verwenden Sie einen Punkt für Dezimalzahlen.

Anführungszeichen im Ergebnis kennzeichnen Text; ein Rückwärtsschrägstrich mit anschließendem n kennzeichnet einen Zeilenumbruch. Datumswerte verwenden die ISO-Schreibweise. Die Darstellung hängt vom Feldtyp und der Oberflächensprache ab.

Ihre Felder verwenden

Ersetzen Sie Beispielwerte durch Ihre Feld-API-Namen. Datensatzformeln verwenden Namen wie `amount`; Flow-Ausdrücke verwenden Kontext wie `$Datensatz.amount`. Wählen Sie einen passenden Rückgabetyp.

ABS

Zurück zum Funktionskatalog

Bedeutung

Liefert den Betrag einer Zahl ohne Vorzeichen.

Syntax

```
ABS(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ABS(-125)
```

Ergebnis

```
125
```

Verhalten und Grenzen

Nützlich für die Größe einer Differenz, unabhängig von ihrer Richtung.

ACOS

Zurück zum Funktionskatalog

Bedeutung

Liefert den Arkuskosinus als Winkel im Bogenmaß.

Syntax

```
ACOS(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
number	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ACOS(1)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Die Eingabe muss zwischen -1 und 1 liegen. Ein numerisches Ergebnis außerhalb des Definitionsbereichs bleibt bei Speicherung als numerisches Formelfeld leer.

ADDMONTHS

Zurück zum Funktionskatalog

Bedeutung

Verschiebt ein Datum um eine Anzahl von Kalendermonaten.

Syntax

```
ADDMONTHS(date_or_datetime, months)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert.
`months`	Anzahl der hinzuzufügenden Monate; negative Werte verschieben rückwärts. Nachkommastellen werden abgeschnitten.

Rückgabe: Datum.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ADDMONTHS(DATE(2026, 1, 31), 1)
```

Ergebnis

```
"2026-02-28"
```

Beispiel 2

Formel

```
ADDMONTHS(DATE(2026, 3, 31), -1)
```

Ergebnis

```
"2026-02-28"
```

Verhalten und Grenzen

Begrenzt den Tag auf den letzten Tag des Zielmonats. Dies ist keine Addition von 30 Tagen.

AND

Zurück zum Funktionskatalog

Bedeutung

Liefert nur dann true, wenn jede Bedingung wahr ist.

Syntax

```
AND(condition1, condition2, ...)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`condition1`	Erste Wahrheitswertbedingung; mindestens eine ist erforderlich.
`condition2, ...`	Optionale weitere Bedingungen.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
AND(10 > 5, 3 < 7)
```

Ergebnis

```
true
```

Verhalten und Grenzen

Bedingungen werden von links nach rechts ausgewertet; sobald das Ergebnis feststeht, endet die Auswertung.

ASCII

Zurück zum Funktionskatalog

Bedeutung

Liefert die UTF-16-Codeeinheit des ersten Zeichens.

Syntax

```
ASCII(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Text, dessen erstes Zeichen untersucht wird.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ASCII("ABC")
```

Ergebnis

```
65
```

Verhalten und Grenzen

Trotz des Namens verwendet die Funktion JavaScript-Zeichencodes. Leerer Text liefert einen Leerwert.

ASIN

Zurück zum Funktionskatalog

Bedeutung

Liefert den Arkussinus als Winkel im Bogenmaß.

Syntax

```
ASIN(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
number	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ASIN(0)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Die Eingabe muss zwischen -1 und 1 liegen.

ATAN

Zurück zum Funktionskatalog

Bedeutung

Liefert den Arkustangens als Winkel im Bogenmaß.

Syntax

```
ATAN(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
number	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ATAN(0)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Verwenden Sie ATAN2 bei getrennten vertikalen und horizontalen Komponenten.

ATAN2

Zurück zum Funktionskatalog

Bedeutung

Liefert den Winkel eines zweidimensionalen Vektors im Bogenmaß.

Syntax

```
ATAN2(y, x)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
y	Vertikale Komponente; dies ist das ERSTE Argument.
x	Horizontale Komponente; dies ist das ZWEITE Argument.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ATAN2(0, 1)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Die Reihenfolge ist y, x. Beide Vorzeichen bestimmen den Quadranten.

ATTACHMENTCOUNT

Zurück zum Funktionskatalog

Bedeutung

Liefert die Anzahl der Anhänge des aktuellen Datensatzes.

Syntax

```
ATTACHMENTCOUNT()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Beispielkontext: Der aktuelle Datensatz hat 3 Anhänge.

Formel

```
ATTACHMENTCOUNT()
```

Ergebnis

```
3
```

Kompatibilität: `ANZAHL_ANHAENGE` bleibt in vorhandenen Formeln gültig. Verwenden Sie `ATTACHMENTCOUNT` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Das Beispiel verwendet einen aktuellen Datensatz mit drei Anhängen. Ein Kontext ohne Anhanganzahl liefert 0 beziehungsweise false; diese Funktionen zählen keine Datei-/Bildfelder.

BEGINS

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein Text mit einer bestimmten Zeichenfolge beginnt.

Syntax

```
BEGINS(text, prefix)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Vollständiger zu prüfender Text.
`prefix`	Am Anfang erwarteter Text.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
BEGINS("INV-1042", "INV-")
```

Ergebnis

```
true
```

Verhalten und Grenzen

Der Vergleich unterscheidet Groß- und Kleinschreibung.

BLANKVALUE

Zurück zum Funktionskatalog

Bedeutung

Ersetzt einen fehlenden Wert oder leeren Text durch einen Ersatzwert.

Syntax

```
BLANKVALUE(value, fallback)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Ursprünglicher Wert, der verwendet wird, wenn er nicht leer ist.
`fallback`	Ersatzwert, wenn der ursprüngliche Wert fehlt oder leer ist.

Rückgabe: Typ des ausgewählten Werts.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
BLANKVALUE("", "Not specified")
```

Ergebnis

```
"Not specified"
```

Beispiel 2

Formel

```
BLANKVALUE(0, 100)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Null als Zahl und false bleiben erhalten. Die beiden Funktionen verhalten sich derzeit identisch.

BR

Zurück zum Funktionskatalog

Bedeutung

Fügt einen Zeilenumbruch in Text ein.

Syntax

```
BR()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
"First line" & BR() & "Second line"
```

Ergebnis

```
"First line\nSecond line"
```

Verhalten und Grenzen

Das Ergebnis ist ein Zeilenumbruchzeichen, kein HTML-Tag ‹br›. Ob es als sichtbarer Umbruch erscheint, hängt vom Ziel ab.

CASE

Zurück zum Funktionskatalog

Bedeutung

Vergleicht einen Wert mit Alternativen und liefert das erste passende Ergebnis.

Syntax

```
CASE(value, match1, result1, match2, result2, ..., fallback)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Zu vergleichender Wert.
`match1, result1`	Erster Vergleichswert, gefolgt von seinem Ergebnis.
`match2, result2, ...`	Optionale weitere Vergleichs-/Ergebnispaare, die der Reihe nach geprüft werden.
`fallback`	Wert für den Fall ohne Treffer; geben Sie ihn ausdrücklich an.

Rückgabe: Typ des ausgewählten Werts.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
CASE("won", "new", 10, "won", 100, 0)
```

Ergebnis

```
100
```

Beispiel 2

Formel

```
CASE("unknown", "won", 100, 0)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Das Register erlaubt mindestens 3 Argumente; ein vollständiger Aufruf mit einem Paar und Ersatzwert hat jedoch 4. Ohne Ersatzwert und ohne Treffer bleibt das Ergebnis leer. Vergleichs-/Ergebnispaare müssen zusammenbleiben.

CASESAFEID

Zurück zum Funktionskatalog

Bedeutung

Liefert die Eingabe als Text, ohne die Kennung umzuwandeln.

Syntax

```
CASESAFEID(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Kennung oder Wert, der in Text umgewandelt werden soll.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
CASESAFEID("DEMO_ACCOUNT")
```

Ergebnis

```
"DEMO_ACCOUNT"
```

Verhalten und Grenzen

Nur Kompatibilitätsverhalten: Es wird weder eine Prüfsumme ergänzt noch eine ID-Prüfung durchgeführt.

CEILING

Zurück zum Funktionskatalog

Bedeutung

Rundet eine Zahl von null weg auf eine ganze Zahl.

Syntax

```
CEILING(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
CEILING(-3.2)
```

Ergebnis

```
-4
```

Verhalten und Grenzen

Dies unterscheidet sich bei negativen Zahlen von MCEILING.

CHR

Zurück zum Funktionskatalog

Bedeutung

Erzeugt ein UTF-16-Zeichen aus einem numerischen Code.

Syntax

```
CHR(code)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
code	Zeichencode; normalerweise eine ganze Zahl von 0 bis 65535.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
CHR(65)
```

Ergebnis

```
"A"
```

Verhalten und Grenzen

Dies ist kein vollständiger Unicode-Codepunkt-Konstruktor für ergänzende Zeichen außerhalb der Basisebene.

CONTAINS

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein Text eine Teilzeichenfolge enthält.

Syntax

```
CONTAINS(text, search_text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu durchsuchender Text.
`search_text`	Gesuchte Teilzeichenfolge.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
CONTAINS("Service planning", "planning")
```

Ergebnis

```
true
```

Verhalten und Grenzen

Der Vergleich unterscheidet Groß- und Kleinschreibung.

COS

Zurück zum Funktionskatalog

Bedeutung

Liefert den Kosinus eines Winkels.

Syntax

```
COS(angle)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`angle`	Winkel im Bogenmaß, nicht in Grad.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
COS(0)
```

Ergebnis

```
1
```

Verhalten und Grenzen

Rechnen Sie Grad mit degrees * PI() / 180 in Bogenmaß um.

CURRENCYRATE

Zurück zum Funktionskatalog

Bedeutung

Liefert Einheiten der Unternehmenswährung je Einheit der angefragten Währung.

Syntax

```
CURRENCYRATE(currency_code)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
currency_code	Dreibuchstabiger ISO-Währungscode, etwa "USD". Groß-/Kleinschreibung spielt keine Rolle. Die Unternehmenswährung ist standardmäßig EUR.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Beispielkontext: Unternehmenswährung EUR; die gezeigten Wechselkurse sind feste fiktive Testwerte.

Formel

```
CURRENCYRATE("USD")
```

Ergebnis

```
0.8
```

Beispiel 2

Beispielkontext: Unternehmenswährung EUR; die gezeigten Wechselkurse sind feste fiktive Testwerte.

Formel

```
100 * CURRENCYRATE("USD")
```

Ergebnis

```
80
```

Beispiel 3

Beispielkontext: Unternehmenswährung EUR; die gezeigten Wechselkurse sind feste fiktive Testwerte.

Formel

```
CURRENCYRATE("EUR")
```

Ergebnis

```
1
```

Verhalten und Grenzen

Die Beispiele verwenden einen festen fiktiven Kurs, keine aktuelle Notierung. Multiplizieren Sie einen Fremdwährungsbetrag mit dem Rückgabewert, um ihn in die Unternehmenswährung umzurechnen. Die konfigurierte kostenlose Frankfurter-Quelle liefert Referenzkurse, keine Live-Handelskurse. Feldwerk prüft bei eintreffenden API-Anfragen höchstens stündlich auf neue Kurse. GET /api/wechselkurse meldet Quelle, ältestes enthaltenes Beobachtungsdatum, Abrufzeit und Ablauf. Bei Anbieterfehlern bleibt der letzte gültige Schnappschuss bis zum Ablauf nutzbar; fehlgeschlagene Abrufe können nach einer Minute erneut versucht werden. Ein Schnappschuss läuft spätestens nach 36 Stunden ab und enthält niemals Beobachtungen älter als sieben Tage. Fehlende, nicht unterstützte oder abgelaufene Kurse liefern null, niemals einen geratenen 1:1-Kurs. Die Unternehmenswährung selbst liefert bei verfügbarem Serverkontext immer 1. Dieses Verhalten bei fehlenden Kursen unterscheidet sich bewusst von Systemen, die für ungültige Codes 1 liefern. Kurse werden beim Lesen ausgewertet; sie fixieren keinen historischen Rechnungswechselkurs.

DATE

Zurück zum Funktionskatalog

Bedeutung

Erzeugt ein Datum aus drei Zahlen.

Syntax

```
DATE(year, month, day)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`year`	Vollständiges Jahr, beispielsweise 2026.
`month`	Monatsnummer, normalerweise 1–12.
`day`	Tag des Monats, normalerweise 1–31.

Rückgabe: Datum.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
DATE(2026, 12, 15)
```

Ergebnis

```
"2026-12-15"
```

Verhalten und Grenzen

Verwenden Sie gültige Kalenderbestandteile. Die aktuelle Engine erzeugt UTC-Daten und normalisiert überlaufende Bestandteile, statt jedes ungültige Kalenderdatum abzulehnen.

DATETIMEVALUE

Zurück zum Funktionskatalog

Bedeutung

Liest einen Zeitstempel ein oder wandelt ein typisiertes Datum in Datum/Uhrzeit um.

Syntax

```
DATETIMEVALUE(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	ISO-Zeitstempel als Text oder ein typisiertes Datum. Ein Leerzeichen zwischen Datum und Uhrzeit wird akzeptiert.

Rückgabe: Datum/Uhrzeit.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
DATETIMEVALUE("2026-12-15 09:30:00")
```

Ergebnis

```
"2026-12-15T09:30:00Z"
```

Verhalten und Grenzen

Text ohne Zeitzonensuffix wird als UTC interpretiert, nicht in der lokalen Zeitzone des Benutzers. Ungültige Zeitstempel liefern einen Leerwert.

DATEVALUE

Zurück zum Funktionskatalog

Bedeutung

Wandelt ISO-Datumstext oder typisiertes Datum/Uhrzeit in ein Datum um.

Syntax

```
DATEVALUE(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
value	Text im Format YYYY-MM-DD oder ein typisierter Datums-/Datum-Uhrzeit-Wert.

Rückgabe: Datum.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
DATEVALUE("2026-12-15")
```

Ergebnis

```
"2026-12-15"
```

Verhalten und Grenzen

Ein typisierter Datum/Uhrzeit-Wert wird auf sein UTC-Datum gekürzt. Übergeben Sie hier keinen vollständigen Zeitstempeltext; lesen Sie ihn zuerst mit DATETIMEVALUE ein.

DAY

Zurück zum Funktionskatalog

Bedeutung

Liest den Tag des Monats aus einem Datum.

Syntax

```
DAY(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
DAY(DATE(2026, 12, 15))
```

Ergebnis

```
15
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

DAYOFYEAR

Zurück zum Funktionskatalog

Bedeutung

Liest die Tagesnummer innerhalb des Jahres (1–366) aus einem Datum.

Syntax

```
DAYOFYEAR(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
DAYOFYEAR(DATE(2026, 2, 1))
```

Ergebnis

```
32
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

DISTANCE

Zurück zum Funktionskatalog

Bedeutung

Berechnet die Großkreisentfernung zwischen zwei Koordinatenpaaren.

Syntax

```
DISTANCE(location1, location2, unit)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`location1`	Mit GEOLOCATION erzeugtes Ausgangskoordinatenpaar.
`location2`	Mit GEOLOCATION erzeugtes Zielkoordinatenpaar.
`unit`	"km" für Kilometer oder "mi" für Meilen.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ROUND(DISTANCE(GEOLOCATION(0, 0), GEOLOCATION(0, 1), "km"), 3)
```

Ergebnis

```
111.195
```

Beispiel 2

Formel

```
ROUND(DISTANCE(GEOLOCATION(0, 0), GEOLOCATION(0, 1), "mi"), 3)
```

Ergebnis

```
69.093
```

Verhalten und Grenzen

Verwendet eine kugelförmige Erde mit mittlerem Radius 6371,0088 km. Dies ist eine angenäherte Oberflächenentfernung, keine Fahrstrecke. Fehlende Koordinaten liefern null. Ungültige Koordinaten oder Einheiten erzeugen einen Fehler.

EXP

Zurück zum Funktionskatalog

Bedeutung

Potenziert die mathematische Konstante e mit dem angegebenen Exponenten.

Syntax

```
EXP(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
EXP(0)
```

Ergebnis

```
1
```

Verhalten und Grenzen

Die Eingabe ist ein Exponent, keine Basis. Verwenden Sie für andere Basen den Operator ^.

FIND

Zurück zum Funktionskatalog

Bedeutung

Findet die erste Position einer Teilzeichenfolge.

Syntax

```
FIND(search_text, text [, start])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`search_text`	Gesuchte Teilzeichenfolge; dieses Argument steht zuerst.
`text`	Vollständiger zu durchsuchender Text.
`start`	Optionale erste Suchposition; Standard ist 1.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FIND("plan", "Service planning")
```

Ergebnis

```
9
```

Beispiel 2

Formel

```
FIND("a", "banana", 3)
```

Ergebnis

```
4
```

Verhalten und Grenzen

Positionen beginnen bei 1. Liefert 0, wenn nichts gefunden wird oder die Startposition außerhalb des Texts liegt.

FIXED

Zurück zum Funktionskatalog

Bedeutung

Formatiert eine Zahl als Text mit festen Dezimalstellen, Dezimalpunkt und ohne Gruppierung.

Syntax

```
FIXED(number [, decimal_places])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Zu formatierender Wert.
`decimal_places`	Optional; Standard ist 2. Wird zur ganzen Zahl gekürzt und auf 0–15 begrenzt.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FIXED(2400, 2)
```

Ergebnis

```
"2400.00"
```

Beispiel 2

Formel

```
FIXED(12)
```

Ergebnis

```
"12.00"
```

Kompatibilität: `FESTKOMMA` bleibt in vorhandenen Formeln gültig. Verwenden Sie `FIXED` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Das Ergebnis ist Text, kein numerischer Betrag. Das Format ist unabhängig von der Oberflächensprache. Eine leere numerische Eingabe liefert leeren Text.

FLOOR

Zurück zum Funktionskatalog

Bedeutung

Rundet eine Zahl in Richtung null auf eine ganze Zahl.

Syntax

```
FLOOR(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FLOOR(-3.8)
```

Ergebnis

```
-3
```

Verhalten und Grenzen

Dies unterscheidet sich bei negativen Zahlen von MFLOOR.

FORMATDURATION

Zurück zum Funktionskatalog

Bedeutung

Formatiert eine verstrichene Dauer als Stunden:Minuten:Sekunden.

Syntax

```
FORMATDURATION(seconds [, compatibility_option])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`seconds`	Dauer in Sekunden. Verwendet wird der absolute ganzzahlige Anteil.
`compatibility_option`	Optionales zweites Argument; wird akzeptiert, aber ignoriert.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FORMATDURATION(3661)
```

Ergebnis

```
"1:01:01"
```

Beispiel 2

Formel

```
FORMATDURATION(-61, true)
```

Ergebnis

```
"0:01:01"
```

Verhalten und Grenzen

Stunden können größer als 23 sein. Die Funktion formatiert keinen Kalenderzeitstempel. Negative Dauern verlieren ihr Vorzeichen.

FORMATNUMBERDE

Zurück zum Funktionskatalog

Bedeutung

Formatiert eine Zahl als deutschen Zahlentext mit Tausendergruppierung und Dezimalkomma.

Syntax

```
FORMATNUMBERDE(number [, decimal_places])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Zu formatierender Wert.
`decimal_places`	Optional; Standard ist 2. Wird zur ganzen Zahl gekürzt und auf 0–15 begrenzt.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FORMATNUMBERDE(2400, 2)
```

Ergebnis

```
"2.400,00"
```

Beispiel 2

Formel

```
FORMATNUMBERDE(12)
```

Ergebnis

```
"12,00"
```

Kompatibilität: `TEXTDE` bleibt in vorhandenen Formeln gültig. Verwenden Sie `FORMATNUMBERDE` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Das Ergebnis ist Text, kein numerischer Betrag. Das Format ist unabhängig von der Oberflächensprache. Eine leere numerische Eingabe liefert leeren Text.

FROMUNIXTIME

Zurück zum Funktionskatalog

Bedeutung

Wandelt Unix-Epochensekunden in Datum/Uhrzeit um.

Syntax

```
FROMUNIXTIME(seconds)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
seconds	Sekunden seit dem 1. Januar 1970, 00:00:00 UTC.

Rückgabe: Datum/Uhrzeit.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
FROMUNIXTIME(60)
```

Ergebnis

```
"1970-01-01T00:01:00Z"
```

Verhalten und Grenzen

Teilen Sie einen JavaScript-Millisekundenzeitstempel vor der Übergabe durch 1000.

GEOLOCATION

Zurück zum Funktionskatalog

Bedeutung

Erzeugt ein Koordinatenpaar für eine DISTANCE-Berechnung.

Syntax

```
GEOLOCATION(latitude, longitude)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`latitude`	Breitengrad in Dezimalgrad zwischen -90 und 90.
`longitude`	Längengrad in Dezimalgrad zwischen -180 und 180.

Rückgabe: Internes Koordinatenpaar (mit DISTANCE verwenden).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ROUND(DISTANCE(GEOLOCATION(0, 0), GEOLOCATION(0, 1), "km"), 3)
```

Ergebnis

```
111.195
```

Verhalten und Grenzen

Das Koordinatenpaar ist ein interner Wert, kein neuer speicherbarer Feldtyp. Verwenden Sie es innerhalb von DISTANCE. Leere Koordinaten liefern null; Koordinaten außerhalb des zulässigen Bereichs erzeugen einen Fehler. Die Funktion sucht keine Adresse und ruft keinen Kartendienst auf.

GETSESSIONID

Zurück zum Funktionskatalog

Bedeutung

Liefert die öffentliche Kennung der aktuellen angemeldeten Browsersitzung.

Syntax

```
GETSESSIONID()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Beispielkontext: eine angemeldete Browsersitzung mit der unten gezeigten fiktiven öffentlichen Kennung. Ohne Sitzungskontext ist das Ergebnis leerer Text.

Formel

```
GETSESSIONID()
```

Ergebnis

```
"SES0123456789ABCDE"
```

Beispiel 2

Formel

```
GETSESSIONID()
```

Ergebnis

```
""
```

Verhalten und Grenzen

Die 18-stellige Kennung verwendet das Präfix SES und dasselbe Alphabet wie Datensatz-IDs. Sie ist zufällig, innerhalb der Sitzung stabil und kann keine Anfrage authentifizieren. Sie ist weder das geheime Anmeldetoken noch dessen Hash. Eine neue Anmeldung erhält eine andere Kennung. API-Token-Anfragen, öffentliche Seiten und geplante Aufgaben ohne Browsersitzung liefern leeren Text. Berechnete Feldwerte können daher zwischen Sitzungen abweichen; verwenden Sie dies nicht als stabile Geschäftskennung.

HASATTACHMENTS

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob der aktuelle Datensatz mindestens einen Anhang hat.

Syntax

```
HASATTACHMENTS()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Beispielkontext: Der aktuelle Datensatz hat 3 Anhänge.

Formel

```
HASATTACHMENTS()
```

Ergebnis

```
true
```

Kompatibilität: `HAT_ANHAENGE` bleibt in vorhandenen Formeln gültig. Verwenden Sie `HASATTACHMENTS` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Das Beispiel verwendet einen aktuellen Datensatz mit drei Anhängen. Ein Kontext ohne Anhanganzahl liefert 0 beziehungsweise false; diese Funktionen zählen keine Datei-/Bildfelder.

HOUR

Zurück zum Funktionskatalog

Bedeutung

Liest die Stunde (0–23) aus einem Datum.

Syntax

```
HOUR(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
HOUR(DATETIMEVALUE("2026-12-15 09:30:45"))
```

Ergebnis

```
9
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

HYPERLINK

Zurück zum Funktionskatalog

Bedeutung

Erzeugt einen Linkwert mit Zieladresse und Anzeigetext.

Syntax

```
HYPERLINK(url, label [, target])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`url`	Zieladresse.
`label`	Für den Link angezeigter Text.
`target`	Optionales Kompatibilitätsargument; wird derzeit ignoriert.

Rückgabe: Link.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
HYPERLINK("https://example.invalid/help", "Help")
```

Ergebnis

```
{"url": "https://example.invalid/help", "text": "Help"}
```

Beispiel 2

Formel

```
HYPERLINK("https://example.invalid/help", "Help", "_blank")
```

Ergebnis

```
{"url": "https://example.invalid/help", "text": "Help"}
```

Verhalten und Grenzen

Wählen Sie Link als Rückgabetyp des Formelfelds. Das Argument target steuert nicht, welcher Browser-Tab geöffnet wird.

IF

Zurück zum Funktionskatalog

Bedeutung

Wählt anhand einer Bedingung zwischen zwei Werten.

Syntax

```
IF(condition, value_if_true, value_if_false)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`condition`	Zuerst auszuwertende Wahrheitswertbedingung.
`value_if_true`	Rückgabewert, wenn die Bedingung wahr ist.
`value_if_false`	Rückgabewert, wenn die Bedingung falsch ist; dieses Argument ist erforderlich.

Rückgabe: Typ des ausgewählten Werts.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
IF(18000 > 10000, "Priority", "Standard")
```

Ergebnis

```
"Priority"
```

Verhalten und Grenzen

Nur der gewählte Zweig wird ausgewertet. Halten Sie beide Wertzweige mit dem gewählten Rückgabetyp des Formelfelds kompatibel.

IMAGE

Zurück zum Funktionskatalog

Bedeutung

Liefert den Alternativtext eines Bilds; diese Implementierung stellt kein Bild dar.

Syntax

```
IMAGE(url, alternative_text [, height, width])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`url`	Bildadresse; wird akzeptiert, aber nicht zum Darstellen eines Bilds verwendet.
`alternative_text`	Von der Funktion zurückgegebener Text.
`height, width`	Optionale Kompatibilitätsmaße; werden ignoriert.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
IMAGE("https://example.invalid/logo.png", "Company logo")
```

Ergebnis

```
"Company logo"
```

Verhalten und Grenzen

Verwenden Sie ein Datei-/Bildfeld und die passende Seitenkomponente, wenn Sie ein Bild anzeigen möchten.

INCLUDES

Zurück zum Funktionskatalog

Bedeutung

Prüft auf einen exakten Eintrag in einem Mehrfachauswahlwert.

Syntax

```
INCLUDES(choices, stored_value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`choices`	Mehrfachauswahlfeld; seine gespeicherten Werte werden als durch Semikolons getrennter Text dargestellt.
`stored_value`	Gesuchter exakter gespeicherter API-Wert, keine Anzeigebeschriftung.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
INCLUDES("email;phone", "phone")
```

Ergebnis

```
true
```

Verhalten und Grenzen

Der Vergleich unterscheidet Groß- und Kleinschreibung. Eine Teilzeichenfolge innerhalb einer Option ist kein Treffer.

INITCAP

Zurück zum Funktionskatalog

Bedeutung

Wandelt Text in Kleinbuchstaben und Wortanfänge in Großbuchstaben um.

Syntax

```
INITCAP(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
INITCAP("SERVICE PLANNING")
```

Ergebnis

```
"Service Planning"
```

Verhalten und Grenzen

Verwendet einfache Wortgrenzen. Prüfen Sie Namen mit Satzzeichen oder Nicht-ASCII-Zeichen.

ISBLANK

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein Wert fehlt oder eine leere Zeichenfolge ist.

Syntax

```
ISBLANK(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
value	Zu untersuchendes Feld oder zu untersuchender Ausdruck.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISBLANK("")
```

Ergebnis

```
true
```

Beispiel 2

Formel

```
ISBLANK(0)
```

Ergebnis

```
false
```

Verhalten und Grenzen

Diese beiden Funktionen verhalten sich derzeit identisch. Die Zahl null, false und Text nur aus Leerzeichen gelten nicht als leer.

ISNULL

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein Wert fehlt oder eine leere Zeichenfolge ist.

Syntax

```
ISNULL(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Zu untersuchendes Feld oder zu untersuchender Ausdruck.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISNULL("")
```

Ergebnis

```
true
```

Beispiel 2

Formel

```
ISNULL(0)
```

Ergebnis

```
false
```

Verhalten und Grenzen

Diese beiden Funktionen verhalten sich derzeit identisch. Die Zahl null, false und Text nur aus Leerzeichen gelten nicht als leer.

ISNUMBER

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein Wert numerisch ist oder als Zahl gelesen werden kann.

Syntax

```
ISNUMBER(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
value	Zu untersuchende Zahl oder zu untersuchender Text.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISNUMBER("125.50")
```

Ergebnis

```
true
```

Beispiel 2

Formel

```
ISNUMBER("workshop")
```

Ergebnis

```
false
```

Verhalten und Grenzen

Akzeptiert numerischen Text, nicht nur Zahlenfelder.

ISOWEEK

Zurück zum Funktionskatalog

Bedeutung

Liest die ISO-Kalenderwochennummer aus einem Datum.

Syntax

```
ISOWEEK(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISOWEEK(DATE(2026, 11, 23))
```

Ergebnis

```
48
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers. Um den Jahreswechsel kann das ISO-Jahr vom Kalenderjahr abweichen.

ISOYEAR

Zurück zum Funktionskatalog

Bedeutung

Liest das ISO-Kalenderwochenjahr aus einem Datum.

Syntax

```
ISOYEAR(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISOYEAR(DATE(2021, 1, 1))
```

Ergebnis

```
2020
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers. Um den Jahreswechsel kann das ISO-Jahr vom Kalenderjahr abweichen.

ISPICKVAL

Zurück zum Funktionskatalog

Bedeutung

Vergleicht eine einfache Auswahlliste mit einem gespeicherten API-Wert.

Syntax

```
ISPICKVAL(picklist, stored_value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`picklist`	Das Auswahllistenfeld oder sein gespeicherter Textwert.
`stored_value`	Exakter gespeicherter API-Wert in Anführungszeichen; nicht seine übersetzte Anzeigebeschriftung.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ISPICKVAL("won", "won")
```

Ergebnis

```
true
```

Verhalten und Grenzen

Der Vergleich unterscheidet Groß- und Kleinschreibung. Verwenden Sie für Mehrfachauswahllisten INCLUDES.

LEFT

Zurück zum Funktionskatalog

Bedeutung

Entnimmt Zeichen vom Textanfang.

Syntax

```
LEFT(text, count)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Quelltext.
`count`	Anzahl beizubehaltender Zeichen; null oder eine negative Anzahl liefert leeren Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LEFT("INV-1042", 3)
```

Ergebnis

```
"INV"
```

Verhalten und Grenzen

Ist die Anzahl größer als die Quelllänge, wird der gesamte Text zurückgegeben.

LEN

Zurück zum Funktionskatalog

Bedeutung

Zählt die Zeichen eines Texts.

Syntax

```
LEN(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LEN("Workshop")
```

Ergebnis

```
8
```

Verhalten und Grenzen

Die Engine zählt UTF-16-Codeeinheiten; manche Emojis zählen als zwei.

LN

Zurück zum Funktionskatalog

Bedeutung

Liefert den natürlichen Logarithmus (Basis e).

Syntax

```
LN(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LN(1)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Verwenden Sie positive Eingaben. Nicht endliche numerische Ergebnisse bleiben in einem numerischen Formelfeld leer.

LOCALTEXT

Zurück zum Funktionskatalog

Bedeutung

Formatiert Datum/Uhrzeit als Text in der Ortszeit Europe/Berlin.

Syntax

```
LOCALTEXT(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; Rohtext gegebenenfalls vorher einlesen.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LOCALTEXT(DATETIMEVALUE("2026-11-23 12:34:56"))
```

Ergebnis

```
"2026-11-23T13:34:56"
```

Kompatibilität: `TEXTLOKAL` bleibt in vorhandenen Formeln gültig. Verwenden Sie `LOCALTEXT` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Sommerzeit wird berücksichtigt. Die Ausgabe hat kein Z-Suffix, da sie Ortszeit und nicht UTC ist.

LOG

Zurück zum Funktionskatalog

Bedeutung

Liefert den Logarithmus zur Basis 10.

Syntax

```
LOG(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LOG(1000)
```

Ergebnis

```
3
```

Verhalten und Grenzen

Akzeptiert ein Argument; ein optionales Argument für die Logarithmusbasis gibt es nicht.

LOWER

Zurück zum Funktionskatalog

Bedeutung

Wandelt Text in Kleinbuchstaben um.

Syntax

```
LOWER(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LOWER("Meadowbridge")
```

Ergebnis

```
"meadowbridge"
```

LPAD

Zurück zum Funktionskatalog

Bedeutung

Füllt Text links bis zur gewünschten Länge auf.

Syntax

```
LPAD(text, length [, padding])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Ursprünglicher Text.
`length`	Gewünschte Gesamtzeichenzahl; verwenden Sie eine nicht negative ganze Zahl.
`padding`	Optionaler Fülltext; Standard ist ein Leerzeichen, das bei Bedarf wiederholt wird.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
LPAD("42", 5, "0")
```

Ergebnis

```
"00042"
```

Beispiel 2

Formel

```
LPAD("A", 3)
```

Ergebnis

```
"  A"
```

Verhalten und Grenzen

Ist der ursprüngliche Text bereits lang genug, wird er auf die gewünschte Länge gekürzt. Mehrzeichiger Fülltext kann abgeschnitten werden.

MAX

Zurück zum Funktionskatalog

Bedeutung

Liefert den größten angegebenen numerischen Wert.

Syntax

```
MAX(number1, number2, ...)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number1`	Erster Wert; mindestens ein Argument ist erforderlich.
`number2, ...`	Optionale weitere Werte als zusätzliche, durch Kommas getrennte Argumente.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MAX(12, 27, 19)
```

Ergebnis

```
27
```

Verhalten und Grenzen

Leere oder nicht numerische Argumente werden ignoriert. Bleibt kein numerischer Wert übrig, ist das Ergebnis leer.

MCEILING

Zurück zum Funktionskatalog

Bedeutung

Rundet in Richtung positiv unendlich auf.

Syntax

```
MCEILING(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MCEILING(-3.2)
```

Ergebnis

```
-3
```

Verhalten und Grenzen

Für eine positive Zahl ist MCEILING(3.2) gleich 4.

MFLOOR

Zurück zum Funktionskatalog

Bedeutung

Rundet in Richtung negativ unendlich ab.

Syntax

```
MFLOOR(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
number	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MFLOOR(-3.2)
```

Ergebnis

```
-4
```

Verhalten und Grenzen

Für eine positive Zahl ist MFLOOR(3.2) gleich 3.

MID

Zurück zum Funktionskatalog

Bedeutung

Entnimmt Zeichen ab einer Position innerhalb eines Texts.

Syntax

```
MID(text, start, count)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Quelltext.
`start`	Erste Zeichenposition, beginnend bei 1.
`count`	Höchstens zurückzugebende Zeichenzahl.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MID("INV-1042", 5, 4)
```

Ergebnis

```
"1042"
```

Verhalten und Grenzen

Ein Beginn hinter dem Text liefert leeren Text. Startwerte unter 1 werden als 1 behandelt.

MILLISECOND

Zurück zum Funktionskatalog

Bedeutung

Liest die Millisekunde (0–999) aus einem Datum.

Syntax

```
MILLISECOND(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MILLISECOND(DATETIMEVALUE("2026-12-15T09:30:45.123Z"))
```

Ergebnis

```
123
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

MIN

Zurück zum Funktionskatalog

Bedeutung

Liefert den kleinsten angegebenen numerischen Wert.

Syntax

```
MIN(number1, number2, ...)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number1`	Erster Wert; mindestens ein Argument ist erforderlich.
`number2, ...`	Optionale weitere Werte als zusätzliche, durch Kommas getrennte Argumente.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MIN(12, 27, 19)
```

Ergebnis

```
12
```

Verhalten und Grenzen

Leere oder nicht numerische Argumente werden ignoriert. Bleibt kein numerischer Wert übrig, ist das Ergebnis leer.

MINUTE

Zurück zum Funktionskatalog

Bedeutung

Liest die Minute (0–59) aus einem Datum.

Syntax

```
MINUTE(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MINUTE(DATETIMEVALUE("2026-12-15 09:30:45"))
```

Ergebnis

```
30
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

MOD

Zurück zum Funktionskatalog

Bedeutung

Liefert den Rest einer Division.

Syntax

```
MOD(dividend, divisor)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`dividend`	Zu teilende Zahl.
`divisor`	Zahl, durch die geteilt wird.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MOD(17, 5)
```

Ergebnis

```
2
```

Beispiel 2

Formel

```
MOD(-17, 5)
```

Ergebnis

```
-2
```

Beispiel 3

Formel

```
MOD(17, 0)
```

Ergebnis

```
17
```

Verhalten und Grenzen

Der Rest folgt dem Vorzeichen des Dividenden. In dieser Implementierung liefert ein Divisor von null den Dividenden zurück.

MONTH

Zurück zum Funktionskatalog

Bedeutung

Liest den Monat (1–12) aus einem Datum.

Syntax

```
MONTH(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
MONTH(DATE(2026, 12, 15))
```

Ergebnis

```
12
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

NOT

Zurück zum Funktionskatalog

Bedeutung

Kehrt eine Wahrheitswertbedingung um.

Syntax

```
NOT(condition)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`condition`	Ein Ausdruck mit dem Ergebnis true oder false.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
NOT(5 > 2)
```

Ergebnis

```
false
```

NOW

Zurück zum Funktionskatalog

Bedeutung

Liefert das aktuelle Datum mit Uhrzeit.

Syntax

```
NOW()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Datum/Uhrzeit.

Durchgerechnete Beispiele

Beispiel 1

Beispielzeitpunkt: 23. November 2026, 12:34:56 UTC.

Formel

```
NOW()
```

Ergebnis

```
"2026-11-23T12:34:56Z"
```

Verhalten und Grenzen

Dieses Beispiel setzt die Uhr auf den 23. November 2026, 12:34:56 UTC. Tatsächliche Ergebnisse hängen vom Auswertungszeitpunkt ab.

NULLVALUE

Zurück zum Funktionskatalog

Bedeutung

Ersetzt einen fehlenden Wert oder leeren Text durch einen Ersatzwert.

Syntax

```
NULLVALUE(value, fallback)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Ursprünglicher Wert, der verwendet wird, wenn er nicht leer ist.
`fallback`	Ersatzwert, wenn der ursprüngliche Wert fehlt oder leer ist.

Rückgabe: Typ des ausgewählten Werts.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
NULLVALUE("", "Not specified")
```

Ergebnis

```
"Not specified"
```

Beispiel 2

Formel

```
NULLVALUE(0, 100)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Null als Zahl und false bleiben erhalten. Die beiden Funktionen verhalten sich derzeit identisch.

OR

Zurück zum Funktionskatalog

Bedeutung

Liefert true, wenn mindestens eine Bedingung wahr ist.

Syntax

```
OR(condition1, condition2, ...)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`condition1`	Erste Wahrheitswertbedingung; mindestens eine ist erforderlich.
`condition2, ...`	Optionale weitere Bedingungen.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
OR(10 < 5, 3 < 7)
```

Ergebnis

```
true
```

Verhalten und Grenzen

Bedingungen werden von links nach rechts ausgewertet; sobald das Ergebnis feststeht, endet die Auswertung.

PI

Zurück zum Funktionskatalog

Bedeutung

Liefert die mathematische Konstante Pi.

Syntax

```
PI()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ROUND(PI(), 4)
```

Ergebnis

```
3.1416
```

Verhalten und Grenzen

Verwenden Sie leere runde Klammern: Diese Funktion hat keine Argumente. Das Beispiel rundet die angenäherte Konstante für die Anzeige.

PICKLISTCOUNT

Zurück zum Funktionskatalog

Bedeutung

Zählt die gewählten Werte einer Mehrfachauswahlliste.

Syntax

```
PICKLISTCOUNT(selection)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`selection`	Ein Mehrfachauswahllistenfeld. In Literalbeispielen werden Optionen durch Semikolons getrennt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
PICKLISTCOUNT("red;blue")
```

Ergebnis

```
2
```

Beispiel 2

Formel

```
PICKLISTCOUNT("")
```

Ergebnis

```
0
```

Verhalten und Grenzen

Leere Optionen werden ignoriert. Ein leeres Feld liefert 0. Verwenden Sie in Datensatzformeln den Feld-API-Namen und in Flow-Ausdrücken $Datensatz.field.

RECORDURL

Zurück zum Funktionskatalog

Bedeutung

Liefert die relative Feldwerk-URL eines referenzierten Datensatzes.

Syntax

```
RECORDURL(lookup)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`lookup`	Eine Nachschlagereferenz mit Zielobjekt und Datensatz-ID, nicht nur ihr Anzeigetext.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Beispielkontext: `account` ist ein Nachschlagefeld, das auf das Objekt `konto`, Datensatz `DEMO_ACCOUNT` (fiktive Kennung), verweist.

Formel

```
RECORDURL(account)
```

Ergebnis

```
"/o/konto/r/DEMO_ACCOUNT"
```

Kompatibilität: `DATENSATZURL` bleibt in vorhandenen Formeln gültig. Verwenden Sie `RECORDURL` in neuen Formeln, unabhängig von der Oberflächensprache.

Verhalten und Grenzen

Das Beispielfeld account verweist auf die unten gezeigte fiktive Firmenreferenz. Leere Eingaben oder Eingaben ohne Referenz liefern leeren Text.

REGEXEXTRACT

Zurück zum Funktionskatalog

Bedeutung

Extrahiert den ersten Treffer eines regulären Ausdrucks oder eine erfasste Gruppe.

Syntax

```
REGEXEXTRACT(text, pattern [, group])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu durchsuchender Text.
`pattern`	Regulärer Ausdruck im JavaScript-Stil als Text in Anführungszeichen, ohne /- Begrenzer.
`group`	Optionale Gruppennummer; 0 (Standard) ist der gesamte Treffer, 1 die erste eingeklammerte Gruppe.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
REGEXEXTRACT("INV-1042", "INV-([0-9]+)", 1)
```

Ergebnis

```
"1042"
```

Beispiel 2

Formel

```
REGEXEXTRACT("INV-1042", "[0-9]+")
```

Ergebnis

```
"1042"
```

Verhalten und Grenzen

Liefert leeren Text, wenn kein Treffer existiert, die Gruppe fehlt oder das Muster ungültig ist.

REGEXTEST

Zurück zum Funktionskatalog

Bedeutung

Prüft, ob ein regulärer Ausdruck auf einen Teil des Texts passt.

Syntax

```
REGEXTEST(text, pattern)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu prüfender Text.
`pattern`	Muster im JavaScript-Stil ohne /-Begrenzer. Mit ^ und $ prüfen Sie den gesamten Wert.

Rückgabe: Wahrheitswert (true/false).

Durchgerechnete Beispiele

Beispiel 1

Formel

```
REGEXTEST("INV-1042", "^INV-[0-9]+$")
```

Ergebnis

```
true
```

Verhalten und Grenzen

Ungültige Muster liefern false. Ein separates Argument für Flags ist nicht verfügbar.

REVERSE

Zurück zum Funktionskatalog

Bedeutung

Kehrt die Reihenfolge der Textzeichen um.

Syntax

```
REVERSE(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
REVERSE("ABC123")
```

Ergebnis

```
"321CBA"
```

Verhalten und Grenzen

Verwendet UTF-16-Codeeinheiten; verlassen Sie sich nicht auf den Erhalt zusammengesetzter Emojis.

RIGHT

Zurück zum Funktionskatalog

Bedeutung

Entnimmt Zeichen vom Textende.

Syntax

```
RIGHT(text, count)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Quelltext.
`count`	Anzahl beizubehaltender Zeichen; null oder eine negative Anzahl liefert leeren Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
RIGHT("INV-1042", 4)
```

Ergebnis

```
"1042"
```

Verhalten und Grenzen

Ist die Anzahl größer als die Quelllänge, wird der gesamte Text zurückgegeben.

ROUND

Zurück zum Funktionskatalog

Bedeutung

Rundet eine Zahl auf die angegebene Anzahl von Dezimalstellen.

Syntax

```
ROUND(number, decimal_places)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Zu rundender Wert.
`decimal_places`	Erforderliche Stellenzahl. 2 bedeutet Hundertstel, 0 eine ganze Zahl, -2 Hunderter.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
ROUND(18.456, 2)
```

Ergebnis

```
18.46
```

Beispiel 2

Formel

```
ROUND(-1.45, 1)
```

Ergebnis

```
-1.5
```

Beispiel 3

Formel

```
ROUND(1250, -2)
```

Ergebnis

```
1300
```

Verhalten und Grenzen

Genau in der Mitte liegende Werte werden von null weg gerundet. Das Ergebnis ist numerisch; verwenden Sie FIXED, wenn nachgestellte Nullen im Text sichtbar sein müssen.

RPAD

Zurück zum Funktionskatalog

Bedeutung

Füllt Text rechts bis zur gewünschten Länge auf.

Syntax

```
RPAD(text, length [, padding])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Ursprünglicher Text.
`length`	Gewünschte Gesamtzeichenzahl; verwenden Sie eine nicht negative ganze Zahl.
`padding`	Optionaler Fülltext; Standard ist ein Leerzeichen, das bei Bedarf wiederholt wird.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
RPAD("A", 4, ".")
```

Ergebnis

```
"A..."
```

Beispiel 2

Formel

```
RPAD("A", 3)
```

Ergebnis

```
"A  "
```

Verhalten und Grenzen

Ist der ursprüngliche Text bereits lang genug, wird er auf die gewünschte Länge gekürzt. Mehrzeichiger Fülltext kann abgeschnitten werden.

SECOND

Zurück zum Funktionskatalog

Bedeutung

Liest die Sekunde (0–59) aus einem Datum.

Syntax

```
SECOND(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
SECOND(DATETIMEVALUE("2026-12-15 09:30:45"))
```

Ergebnis

```
45
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

SIN

Zurück zum Funktionskatalog

Bedeutung

Liefert den Sinus eines Winkels.

Syntax

```
SIN(angle)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`angle`	Winkel im Bogenmaß, nicht in Grad.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
SIN(0)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Rechnen Sie Grad mit degrees * PI() / 180 in Bogenmaß um.

SQRT

Zurück zum Funktionskatalog

Bedeutung

Liefert die nicht negative Quadratwurzel.

Syntax

```
SQRT(number)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Die zu verarbeitende Zahl. Verwenden Sie in Zahlenliteralen einen Dezimalpunkt.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
SQRT(81)
```

Ergebnis

```
9
```

Verhalten und Grenzen

Eine negative Eingabe hat kein endliches reelles Ergebnis und bleibt in einem numerischen Formelfeld leer.

SUBSTITUTE

Zurück zum Funktionskatalog

Bedeutung

Ersetzt jedes Vorkommen einer Teilzeichenfolge.

Syntax

```
SUBSTITUTE(text, old_text, new_text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Ursprünglicher Text.
`old_text`	Exakter Suchtext; kein regulärer Ausdruck.
`new_text`	Ersatztext; verwenden Sie "", um den Treffer zu entfernen.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
SUBSTITUTE("draft-draft", "draft", "final")
```

Ergebnis

```
"final-final"
```

Verhalten und Grenzen

Die Suche unterscheidet Groß- und Kleinschreibung. Ein viertes Argument zur Auswahl eines einzelnen Vorkommens gibt es nicht.

TAN

Zurück zum Funktionskatalog

Bedeutung

Liefert den Tangens eines Winkels.

Syntax

```
TAN(angle)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`angle`	Winkel im Bogenmaß, nicht in Grad.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
TAN(0)
```

Ergebnis

```
0
```

Verhalten und Grenzen

Rechnen Sie Grad mit degrees * PI() / 180 in Bogenmaß um.

TEXT

Zurück zum Funktionskatalog

Bedeutung

Wandelt einen Wert in Text um.

Syntax

```
TEXT(value)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`value`	Zahl, Wahrheitswert, Datum, Datum/Uhrzeit oder Text zur Umwandlung.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
TEXT(1250)
```

Ergebnis

```
"1250"
```

Verhalten und Grenzen

Datum/Uhrzeit wird in UTC mit Z-Suffix umgewandelt. Fehlende Werte werden zu leerem Text. Ein eigener Formatierungsstring wird nicht angewendet.

TIMENOW

Zurück zum Funktionskatalog

Bedeutung

Liefert das aktuelle Datum mit Uhrzeit, in dieser Implementierung wie NOW.

Syntax

```
TIMENOW()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Datum/Uhrzeit.

Durchgerechnete Beispiele

Beispiel 1

Beispielzeitpunkt: 23. November 2026, 12:34:56 UTC.

Formel

```
TIMENOW()
```

Ergebnis

```
"2026-11-23T12:34:56Z"
```

Verhalten und Grenzen

Dieses Beispiel setzt die Uhr auf den 23. November 2026, 12:34:56 UTC. Tatsächliche Ergebnisse hängen vom Auswertungszeitpunkt ab. TIMENOW liefert nicht ausschließlich eine Uhrzeit.

TIMEVALUE

Zurück zum Funktionskatalog

Bedeutung

Liefert aus einem vorhandenen typisierten Datum oder Datum/Uhrzeit einen typisierten Datum/Uhrzeit-Wert.

Syntax

```
TIMEVALUE(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisiertes Datum oder Datum/Uhrzeit, keine rohe Uhrzeitzeichenfolge.

Rückgabe: Datum/Uhrzeit.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
TIMEVALUE(DATETIMEVALUE("2026-12-15 09:30:00"))
```

Ergebnis

```
"2026-12-15T09:30:00Z"
```

Verhalten und Grenzen

Der Datumsanteil bleibt erhalten. TIMEVALUE("09:30:00") liefert in dieser Implementierung einen Leerwert.

TODAY

Zurück zum Funktionskatalog

Bedeutung

Liefert das heutige Datum in UTC.

Syntax

```
TODAY()
```

Argumente in ihrer Reihenfolge

Keine Argumente. Behalten Sie die leeren runden Klammern `()`.

Rückgabe: Datum.

Durchgerechnete Beispiele

Beispiel 1

Beispielzeitpunkt: 23. November 2026, 12:34:56 UTC.

Formel

```
TODAY()
```

Ergebnis

```
"2026-11-23"
```

Verhalten und Grenzen

Dieses Beispiel setzt die Uhr auf den 23. November 2026, 12:34:56 UTC. Tatsächliche Ergebnisse hängen vom Auswertungszeitpunkt ab.

TRIM

Zurück zum Funktionskatalog

Bedeutung

Entfernt Leerraum an beiden Textenden.

Syntax

```
TRIM(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
TRIM("  Service planning  ")
```

Ergebnis

```
"Service planning"
```

Verhalten und Grenzen

Leerzeichen innerhalb des Texts bleiben erhalten.

TRUNC

Zurück zum Funktionskatalog

Bedeutung

Schneidet Dezimalstellen ohne Rundung ab.

Syntax

```
TRUNC(number [, decimal_places])
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`number`	Zu kürzender Wert.
`decimal_places`	Optional; Standard ist 0. Ein positiver Wert behält entsprechend viele Nachkommastellen.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
TRUNC(18.459, 2)
```

Ergebnis

```
18.45
```

Beispiel 2

Formel

```
TRUNC(-18.459)
```

Ergebnis

```
-18
```

Verhalten und Grenzen

Das Abschneiden erfolgt in Richtung null.

UNIXTIMESTAMP

Zurück zum Funktionskatalog

Bedeutung

Wandelt Datum/Uhrzeit in ganze Sekunden seit der Unix-Epoche um.

Syntax

```
UNIXTIMESTAMP(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
UNIXTIMESTAMP(DATETIMEVALUE("1970-01-01 00:01:00"))
```

Ergebnis

```
60
```

Verhalten und Grenzen

Verwendet Sekunden, keine Millisekunden. Sekundenbruchteile werden abgerundet.

UPPER

Zurück zum Funktionskatalog

Bedeutung

Wandelt Text in Großbuchstaben um.

Syntax

```
UPPER(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Zu verarbeitender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
UPPER("Meadowbridge")
```

Ergebnis

```
"MEADOWBRIDGE"
```

URLENCODE

Zurück zum Funktionskatalog

Bedeutung

Kodiert einen einzelnen URL-Bestandteil.

Syntax

```
URLENCODE(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Beispielsweise der Wert eines URL-Abfrageparameters.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
URLENCODE("Service planning & review")
```

Ergebnis

```
"Service%20planning%20%26%20review"
```

Verhalten und Grenzen

Kodieren Sie nicht die gesamte URL, sonst werden Trennzeichen wie / und ? ebenfalls kodiert.

VALUE

Zurück zum Funktionskatalog

Bedeutung

Liest numerischen Text als Zahl ein.

Syntax

```
VALUE(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Text mit einer Zahl, der mit dem gemeinsamen Zahlenparser gelesen wird.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
VALUE("125.50")
```

Ergebnis

```
125.5
```

Verhalten und Grenzen

Ungültiger numerischer Text liefert einen Leerwert. Bevorzugen Sie beim Datenaustausch eindeutige maschinenformatierte Werte.

WEEKDAY

Zurück zum Funktionskatalog

Bedeutung

Liest den Wochentag (1 = Sonntag, 7 = Samstag) aus einem Datum.

Syntax

```
WEEKDAY(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
WEEKDAY(DATE(2026, 11, 23))
```

Ergebnis

```
2
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.

XMLENCODE

Zurück zum Funktionskatalog

Bedeutung

Maskiert XML-relevante Zeichen in einem Textwert.

Syntax

```
XMLENCODE(text)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`text`	Als XML-Inhalt oder Attributwert einzufügender Text.

Rückgabe: Text.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
XMLENCODE("A&B <C>")
```

Ergebnis

```
"A&amp;B &lt;C&gt;"
```

Verhalten und Grenzen

Maskiert &, <, >, doppelte Anführungszeichen und Apostrophe. Wenden Sie die Funktion auf Werte an, nicht auf ein vollständiges XML-Dokument.

YEAR

Zurück zum Funktionskatalog

Bedeutung

Liest das Jahr aus einem Datum.

Syntax

```
YEAR(date_or_datetime)
```

Argumente in ihrer Reihenfolge

Argument	Bedeutung
`date_or_datetime`	Ein typisierter Datums- oder Datum/Uhrzeit-Wert; verwenden Sie DATEVALUE oder DATETIMEVALUE für Rohtext.

Rückgabe: Zahl.

Durchgerechnete Beispiele

Beispiel 1

Formel

```
YEAR(DATE(2026, 12, 15))
```

Ergebnis

```
2026
```

Verhalten und Grenzen

Verwendet UTC-Bestandteile, nicht die Zeitzone des Browsers.