TOPIX®
BUSINESS SOFTWARE AG

DOKU

# TOPIX REST API
# Anwenderbeispiele




Version 26 | Stand 10/2026

enjoy working

Copyright © 1987 – 2026 TOPIX Business Software AG. Alle Rechte bleiben vorbehalten. Alle Inhalte dieser Dokumentation sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen schriftlichen Zustimmung der Autoren.

4D, 4D Write Pro, Apple, Microsoft, Google und weitere genannte Marken sind Eigentum der jeweiligen Markeninhaber und werden anerkannt.

Die enthaltenen Informationen wurden nach bestem Wissen zusammengestellt. Eine Haftung für Vollständigkeit oder Richtigkeit wird nicht übernommen. Diese Dokumentation steht ausschließlich Kunden der TOPIX Business Software AG zur Verfügung.

# Inhaltsverzeichnis


**1 Voraussetzungen 5**

**1.1** Beispielwerte festlegen 5

**1.2** Allgemeine HTTP-Vorgaben 5

**1.3** Mandant und Mandantennummer in TOPIX finden 5

**1.4** SSL-Port in TOPIX finden 6

**2 Authentifizierung 8**

**2.1** Zugangsdaten mit testlogin prüfen 8

**2.2** Bearer Token als Alternative 8

**2.3** Optionale REST-Session erzeugen 9

**3 Kundendaten lesen 10**

**3.1** Kundenliste mit GET abrufen 10

**3.2** Einzelnen Kunden über UUID lesen 10

**4 Kunden anlegen oder aktualisieren 11**

**4.1** Payload aufbauen 11

**4.2** Kunden mit POST senden 12

**4.3** POST mit Bearer Auth 12

**5 Fehlerbehandlung und Sicherheit 13**

**5.1** Wichtige HTTP-Statuscodes 13

**5.2** Sicherheitsregeln 13

**6 Verfügbare HTTP-Methoden 14**

**6.1** GET zum Lesen von Daten 14

**6.2** POST zum Anlegen oder Verarbeiten 15

**6.3** PATCH zum teilweisen Aktualisieren 16

**6.4** DELETE zum Löschen 17

**6.5** GET-only-Bereiche 17

# Willkommen

Diese Anleitung zeigt, wie Sie die TOPIX REST API aus einem externen Programm oder über die Kommandozeile ansprechen. Sie prüfen zuerst die Zugangsdaten und lesen danach Kundendaten mit GET. Anschließend legen Sie einen Kunden mit POST an oder aktualisieren ihn. Die Beispiele verwenden curl, weil curl unter macOS, Linux und Windows verfügbar ist.

# 1 Voraussetzungen

**API-Adresse und Zugangsdaten vorbereiten**

## 1.1 Beispielwerte festlegen

Ersetzen Sie die Beispieladresse durch die URL Ihrer TOPIX-Installation. Der Pfad /restapi/v1 bildet die Basis der in dieser Anleitung verwendeten API-Version. Verwenden Sie eine TOPIX-Mandantennummer und ein Benutzerkonto mit den erforderlichen Rechten.

**Hinweis** Übertragen Sie Zugangsdaten ausschließlich über HTTPS. Speichern Sie Passwörter nicht direkt in Skripten oder in einer Versionsverwaltung.

**Beispiel**

```
Beispiel für die verwendeten Variablen:

BASE_URL=https://topix.example.invalid/restapi/v1
TENANT=1
TOPIX_USER=api-benutzer
TOPIX_PASSWORD=<Passwort des API-Benutzers>

Die Domain example.invalid dient nur als Platzhalter. Ersetzen Sie
sie durch Ihre tatsächliche TOPIX-Adresse.
```

## 1.2 Allgemeine HTTP-Vorgaben

Setzen Sie bei jeder Anfrage den Header Content-Type auf application/json. Verwenden Sie für Antworten das Format application/json.

**Beispiel** Standard-Header:

Content-Type: application/json
Accept: application/json

## 1.3 Mandant und Mandantennummer in TOPIX finden

Lesen Sie den Mandantennamen und die Mandantennummer in TOPIX unter Einstellungen bearbeiten ab. Beide Angaben benötigen Sie für den Pfadparameter tenant. Der Mandant bezeichnet die TOPIX-Datenumgebung, in der die REST-Anfrage arbeitet.

1. Öffnen Sie in TOPIX die Einstellungen.
2. Wählen Sie den Bereich Allgemein.
3. Lesen Sie im Abschnitt Mandant den Namen im Feld Firma 1 ab.
4. Lesen Sie die markierte Mandantennummer im Feld rechts neben Firma 1 ab.

5. Übernehmen Sie die Mandantennummer in den REST-Pfad als tenant.

**Hinweis** Die OpenAPI-Spezifikation beschreibt tenant als Integer. Wenn Ihre Mandantennummer führende Nullen enthält, zum Beispiel 00, stimmen Sie die Schreibweise für Ihre TOPIX-Installation mit der zuständigen Administration ab.

**Beispiel** Beispiel aus dem Screenshot:

Mandant: TOPIX Business Software AG
Mandantennummer: 00

Daraus ergibt sich beispielsweise:

/mandanten/00/firmen

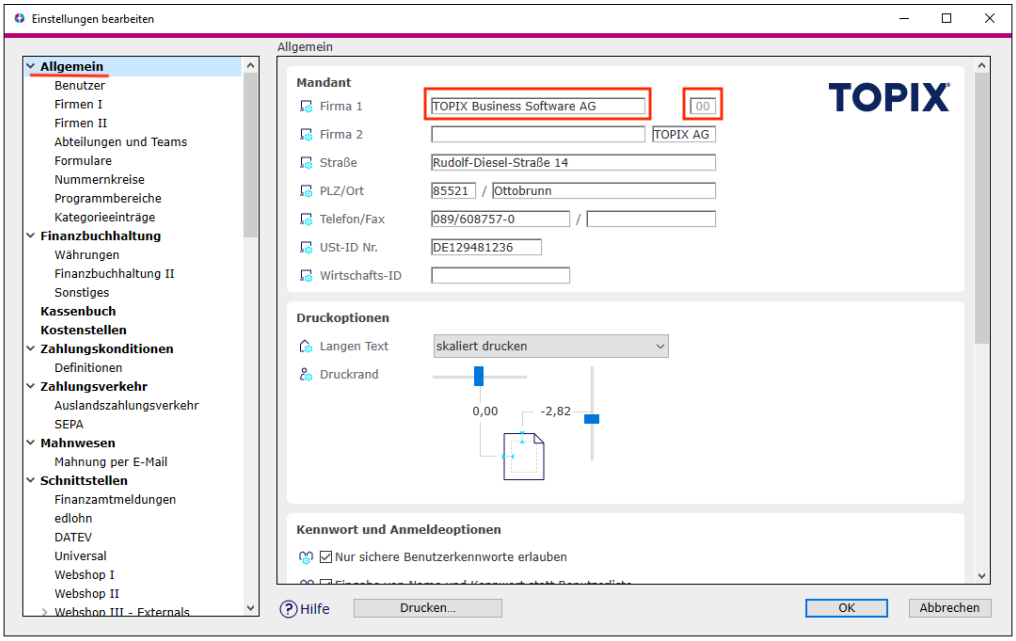



Abbildung 1: Mandant und Mandantennummer im Bereich Allgemein.

## 1.4 SSL-Port in TOPIX finden

Lesen Sie den Port für verschlüsselte HTTPS-Verbindungen in den Kommunikationseinstellungen ab. Verwenden Sie den SSL-Port für die REST-API, wenn Sie die API über HTTPS aufrufen.

1. Öffnen Sie in TOPIX die Einstellungen.
2. Erweitern Sie den Bereich Kommunikation.

3. Wählen Sie den Bereich Internet I.
4. Lesen Sie im Abschnitt Webserver den Wert im Feld SSL-Port ab.
5. Verwenden Sie diesen Port zusammen mit dem Servernamen oder der IP-Adresse in der HTTPS-Basisadresse.

**Tipp**

Der im Screenshot markierte SSL-Port ist 443. Wenn Ihre Installation einen anderen Port verwendet, übernehmen Sie den in TOPIX angezeigten Wert.

**Beispiel**

Beispiel aus dem Screenshot:

SSL-Port: 443

Eine mögliche Basisadresse lautet:

https://<servername>:443/restapi/v1

Ersetzen Sie <servername> durch den Hostnamen oder die IP-Adresse Ihrer TOPIX-Installation.

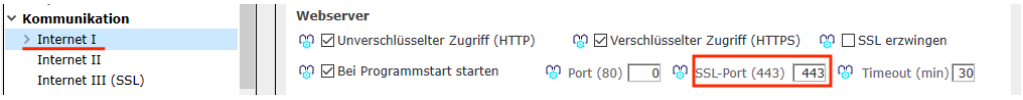



Abbildung 2: SSL-Port im Bereich Kommunikation > Internet I.

# 2 Authentifizierung

**Zugangsdaten prüfen und API-Aufrufe absichern**

## 2.1 Zugangsdaten mit testlogin prüfen

Rufen Sie vor dem ersten fachlichen API-Aufruf den Authentifizierungs-Endpunkt auf. Der Endpunkt prüft die Zugangsdaten und liefert ACL- und Profildaten zurück. Die Prüfung stellt keine dauerhafte Anmeldung im Browser her. Senden Sie die Authentifizierung auch bei den nachfolgenden GET- und POST-Anfragen mit.

1. Setzen Sie BASE_URL, TENANT, TOPIX_USER und TOPIX_PASSWORD.
2. Rufen Sie /mandanten/{tenant}/testlogin mit Basic Auth auf.
3. Prüfen Sie den HTTP-Status sowie die Felder code und message in der JSON-Antwort.
4. Fahren Sie nur bei erfolgreicher Prüfung mit dem eigentlichen API-Aufruf fort.

**Hinweis** Ein HTTP-Status 401 bedeutet, dass die Authentifizierung fehlgeschlagen ist. Ein HTTP-Status 403 weist auf fehlende Berechtigungen oder eine gesperrte Funktion hin.

**Beispiel**

```
curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/testlogin"
```

## 2.2 Bearer Token als Alternative

Die aktuelle Spezifikation unterstützt neben Basic Auth auch Bearer Auth mit einem opaken Token. Verwenden Sie entweder Basic Auth oder Bearer Auth. Mischen Sie beide Verfahren nicht in derselben Anfrage.

**Tipp** Verwenden Sie Bearer Auth nur mit einem Token, das Ihre TOPIX-Umgebung ausstellt oder akzeptiert. Die REST-API-Dokumentation beschreibt nicht, wie ein solcher Token erzeugt wird.

**Beispiel**

```
curl -sS \
  -H "Authorization: Bearer $TOPIX_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/testlogin"
```

## 2.3 Optionale REST-Session erzeugen

Zusätzlich bietet die API einen Endpunkt zum Erzeugen einer REST-Session. Der Request benötigt keine Datensätze. Der Body darf deshalb leer sein; im Beispiel wird ein leeres JSON-Objekt gesendet. Die API-Dokumentation beschreibt die Rückgabe als ApiEnvelope und legt nicht fest, über welchen Header eine erzeugte Session bei späteren Aufrufen übergeben wird. Klären Sie die konkrete Session-Verwendung deshalb für Ihre TOPIX-Version mit dem Systemadministrator.

**Hinweis**

Für die folgenden GET- und POST-Beispiele verwenden Sie weiterhin die dokumentierte Basic- oder Bearer-Authentifizierung. Eine automatische Session-Weitergabe darf nicht angenommen werden.

**Beispiel**

```
Session anlegen:

curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -d '{}' \
  "$BASE_URL/mandanten/$TENANT/sessions"

Session beenden, sobald die UUID aus der Antwort bekannt ist:

curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" -X DELETE \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/sessions/$SESSION_UUID"
```

# 3 Kundendaten lesen

**GET-Endpunkte für Firmen und einzelne Kunden**

## 3.1 Kundenliste mit GET abrufen

Verwenden Sie den Endpunkt /mandanten/{tenant}/firmen, um eine Kunden- beziehungsweise Firmenliste abzurufen. Mit limit und offset begrenzen Sie die Ergebnismenge. Der Parameter detail ist optional und als Integer dokumentiert. Die fachliche Bedeutung der einzelnen Detailstufen muss in Ihrer TOPIX-Umgebung festgelegt werden.

1. Rufen Sie zuerst testlogin auf und prüfen Sie die Antwort.
2. Setzen Sie limit=10 und offset=0, um die ersten zehn Datensätze anzufordern.
3. Senden Sie Basic Auth oder Bearer Auth zusammen mit den JSON-Headern.
4. Lesen Sie die Datensätze im Feld result aus.
5. Verwenden Sie _first, _total und _sent, wenn Sie die Paging-Informationen auswerten.

**Tipp**

Verwenden Sie für produktive Synchronisationen immer limit und offset. So vermeiden Sie unnötig große Antworten und können die Verarbeitung abschnittsweise durchführen.

**Beispiel**

```
curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/firmen?limit=10&offset=0&detail=1"
```

## 3.2 Einzelnen Kunden über UUID lesen

Wenn Sie die UUID eines Kunden kennen, rufen Sie den einzelnen Datensatz über /firmen/{uuid} ab. Ersetzen Sie CUSTOMER_UUID durch die tatsächliche UUID aus einer vorherigen Antwort.

**Hinweis**

Die UUID im Beispiel ist synthetisch. Verwenden Sie nur eine UUID, die in Ihrem TOPIX-Mandanten tatsächlich existiert.

**Beispiel**

```
CUSTOMER_UUID=7f4d6f5e-1111-2222-3333-abcdefabcdef

curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/firmen/$CUSTOMER_UUID"
```

# 4 Kunden anlegen oder aktualisieren

**POST /firmen mit dem data-Wrapper**

## 4.1 Payload aufbauen

Die aktuelle Dokumentation verlangt bei POST /mandanten/{tenant}/firmen ein Top-Level-Feld data. Legen Sie die Kundendaten innerhalb dieses Feldes ab. Der Wrapper data ist verpflichtend. Das Beispiel enthält typische Kunden- und Adressfelder; welche Felder tatsächlich erforderlich sind, hängt von Ihrer TOPIX-Konfiguration und der gewünschten Operation ab.

**Hinweis** Verwenden Sie in der Anfrage echte UTF-8-Feldnamen. Prüfen Sie besonders Felder mit Umlauten oder ß in Ihrer konkreten API-Version.

**Beispiel** Beispiel für kunde.json:

```
{
  "data": {
    "SUCHNAME": "MUSTER",
    "FIRMA1": "Muster GmbH",
    "KD_STATUS": "K",
    "KUERZEL": "MU",
    "USTID_NR": "DE123456789",
    "SPRACHKENNUNG": "de",
    "ANSCHRIFTSDATEN_ARRAY": [
      {
        "ANSCHRIFT_BEZ": "Hauptanschrift",
        "FIRMA1": "Muster GmbH",
        "STRASSE": "Beispielstraße",
        "STRASSE_NR": "1",
        "PLZ1": "80331",
        "ORT": "München",
        "LKZ": "DE",
        "TEL1": "+49 89 123456",
        "E_MAIL": "info@example.invalid"
      }
    ]
  }
}
```

## 4.2 Kunden mit POST senden

Senden Sie die JSON-Datei mit POST an den Firmen-Endpunkt. Verwenden Sie dieselbe Authentifizierung wie beim testlogin.

1. Speichern Sie die Payload als kunde.json.
2. Prüfen Sie, dass data auf oberster Ebene vorhanden ist.
3. Senden Sie die Datei mit Content-Type application/json.
4. Prüfen Sie den HTTP-Status 200 oder 201 sowie die JSON-Antwort.
5. Lesen Sie bei Fehlern code, message und location aus der Antwort aus.

**Tipp**

Die OpenAPI-Spezifikation beschreibt für POST /firmen die Statuscodes 200 für eine erfolgreiche Verarbeitung und 201 für eine Neuanlage.

**Beispiel**

```
curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  --data-binary @kunde.json \
  "$BASE_URL/mandanten/$TENANT/firmen"
```

## 4.3 POST mit Bearer Auth

Wenn Sie Bearer Auth verwenden, ersetzen Sie die Basic-Auth-Option durch den Authorization-Header.

**Beispiel**

```
curl -sS -X POST \
  -H "Authorization: Bearer $TOPIX_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  --data-binary @kunde.json \
  "$BASE_URL/mandanten/$TENANT/firmen"
```

# 5 Fehlerbehandlung und Sicherheit

**Antworten prüfen und Zugangsdaten schützen**

## 5.1 Wichtige HTTP-Statuscodes

Werten Sie bei jeder Anfrage den HTTP-Status und die JSON-Antwort aus. Die API verwendet ein ApiEnvelope mit den Feldern code, message und location; je nach Endpunkt kann zusätzlich result enthalten sein.

**Beispiel** Typische Statuscodes:

200 Anfrage erfolgreich verarbeitet
201 Datensatz oder Session angelegt
400 Ungültige Anfrage
401 Nicht authentifiziert
403 Nicht berechtigt
404 Ressource nicht gefunden
409 Konflikt
415 Content-Type wird nicht unterstützt
422 Fachlich nicht verarbeitbare Daten
423 Ressource gesperrt
501 Funktion für diesen Betriebsmodus nicht implementiert

## 5.2 Sicherheitsregeln

Beachten Sie beim Betrieb einer Integration die folgenden Regeln.

1. Verwenden Sie ausschließlich HTTPS.
2. Legen Sie für die Integration ein eigenes TOPIX-Benutzerkonto mit den minimal erforderlichen Rechten an.
3. Speichern Sie Passwörter und Tokens außerhalb des Quellcodes, zum Beispiel in geschützten Umgebungsvariablen.
4. Protokollieren Sie keine Authorization-Header und keine vollständigen Zugangsdaten.
5. Prüfen Sie vor POST-Aufrufen, ob die übertragene Payload tatsächlich für den vorgesehenen Mandanten bestimmt ist.

**Hinweis** Testen Sie POST-Aufrufe zunächst in einer Testumgebung. Je nach vorhandener UUID oder Payload kann TOPIX einen Datensatz aktualisieren oder neu anlegen.

# 6 Verfügbare HTTP-Methoden

**GET, POST und DELETE nach API-Bereich**

## 6.1 GET zum Lesen von Daten

Verwenden Sie GET, um Daten zu lesen. Die aktuelle OpenAPI-Dokumentation enthält 520 GET-Pfade: 260 Listen-Endpunkte, 259 Einzelzugriffe über UUID sowie den Endpunkt zur Mandantenauswahl. Bei Listen-Endpunkten können je nach Bereich Filter-, Sortier- und Paging-Parameter verfügbar sein.

- GET /mandanten – verfügbare Mandanten ermitteln.
- GET /mandanten/{tenant}/{bereich} – Datensätze eines Bereichs als Liste lesen.
- GET /mandanten/{tenant}/{bereich}/{uuid} – einen einzelnen Datensatz über seine UUID lesen.
- GET mit POST-Unterstützung: angebote, auftraege, firmen, leistungen, personen, tagesabrechnung, terminaldaten, termine, urlaubs-genehmigungen und urlaubsantraege.
- GET mit POST- und PATCH-Unterstützung: angebote, auftraege, firmen, leistungen, personen und termine.
- GET mit DELETE-Unterstützung: dokumente, leistungen, terminaldaten, termine und urlaubsantraege.

**Tipp**

Nicht jeder Listenbereich besitzt zusätzlich einen Einzelzugriff über UUID. Prüfen Sie deshalb den konkreten Pfad in der API-Dokumentation.

**Beispiel**

```
GET-Beispiel: Kundenliste abrufen

Voraussetzung: testlogin wurde erfolgreich ausgeführt.

curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  "$BASE_URL/mandanten/$TENANT/firmen?limit=10&offset=0&detail=1"

Die URL fordert die ersten zehn Firmendatensätze des Mandanten an.
result enthält die Datensätze. Die Felder _first, _total und _sent
liefern Paging-Informationen.
```

**Beispiel** Beispielhafte Antwortstruktur:


```
{
  "code": 0,
  "message": "OK",
  "location": "",
  "_first": 0,
  "_total": 125,
  "_sent": 10,
  "result": [
    {
      "UUID": "<UUID des Kunden>",
      "FIRMA1": "Muster GmbH",
      "SUCHNAME": "MUSTER"
    }
  ]
}
```


Die konkreten Felder in result hängen vom Endpunkt, den Berechtigungen und den Einstellungen des Mandanten ab.

## 6.2 POST zum Anlegen oder Verarbeiten

POST steht in der aktuellen Spezifikation für die folgenden 14 Pfade zur Verfügung. Bei Datensatz-Endpunkten beschreibt POST je nach Ressource das Anlegen oder Aktualisieren. Die Session-Route erzeugt eine REST-Session und verarbeitet keine Datensatzdaten.

- POST /mandanten/{tenant}/angebote
- POST /mandanten/{tenant}/arbeitsberichtsend
- POST /mandanten/{tenant}/auftraege
- POST /mandanten/{tenant}/firmen
- POST /mandanten/{tenant}/leistungen
- POST /mandanten/{tenant}/lieferscheinunterschrift
- POST /mandanten/{tenant}/personen
- POST /mandanten/{tenant}/sessions
- POST /mandanten/{tenant}/tagesabrechnung
- POST /mandanten/{tenant}/terminaldaten
- POST /mandanten/{tenant}/termine

- POST /mandanten/{tenant}/ticketsunterschrift
- POST /mandanten/{tenant}/urlaubs-genehmigungen
- POST /mandanten/{tenant}/urlaubsantraege

**Hinweis** Die in diesem Kapitel aufgeführten Methoden stammen aus der OpenAPI-Spezifikation. Prüfen Sie vor einer produktiven Integration, ob die Methoden in Ihrer installierten TOPIX-Version freigeschaltet sind.

**Beispiel** Für den Kundenbereich verwenden Sie beispielsweise:

POST /mandanten/{tenant}/firmen

Die Nutzdaten müssen gemäß der aktuellen Dokumentation in einem Top-Level-Feld data stehen.

## 6.3 PATCH zum teilweisen Aktualisieren

PATCH aktualisiert einen bestehenden Datensatz über seine UUID. Übertragen Sie nur die Felder, die Sie ändern möchten. Die aktuelle Spezifikation dokumentiert PATCH für Angebote, Aufträge, Firmen, Leistungen, Personen und Termine. Der Request-Body ist verpflichtend und verwendet wie POST den Top-Level-Wrapper data.

- PATCH /mandanten/{tenant}/angebote/{uuid}
- PATCH /mandanten/{tenant}/auftraege/{uuid}
- PATCH /mandanten/{tenant}/firmen/{uuid}
- PATCH /mandanten/{tenant}/leistungen/{uuid}
- PATCH /mandanten/{tenant}/personen/{uuid}
- PATCH /mandanten/{tenant}/termine/{uuid}

**Tipp** Für PATCH ist in der aktuellen Spezifikation bei Angebote, Aufträge, Firmen, Personen und Termine Basic Auth dokumentiert. Für Leistungen sind Basic Auth oder Bearer Auth dokumentiert.

**Beispiel**

```
PATCH-Beispiel: Kundennamen aktualisieren

CUSTOMER_UUID=<UUID des bestehenden Kunden>

curl -sS -u "$TOPIX_USER:$TOPIX_PASSWORD" -X PATCH \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -d '{
    "data": {
      "FIRMA1": "Muster GmbH Neu"
    }
  }' \
  "$BASE_URL/mandanten/$TENANT/firmen/$CUSTOMER_UUID"

Senden Sie nur die Felder, die Sie ändern möchten.
```

## 6.4 DELETE zum Löschen

DELETE ist in der aktuellen Spezifikation für sieben Pfade dokumentiert. Alle DELETE-Pfade erwarten eine UUID des zu löschenden oder zu beendenden Objekts.

- DELETE /mandanten/{tenant}/dokumente/{uuid}
- DELETE /mandanten/{tenant}/leistungen/{uuid}
- DELETE /mandanten/{tenant}/lieferscheinepos/{uuid}
- DELETE /mandanten/{tenant}/sessions/{uuid}
- DELETE /mandanten/{tenant}/terminaldaten/{uuid}
- DELETE /mandanten/{tenant}/termine/{uuid}
- DELETE /mandanten/{tenant}/urlaubsantraege/{uuid}

**Hinweis**

Setzen Sie DELETE nur ein, wenn Sie die fachliche Wirkung und die erforderlichen Berechtigungen geprüft haben. Testen Sie Löschvorgänge zuerst in einer Testumgebung.

## 6.5 GET-only-Bereiche

Die folgenden 249 Bereiche sind in der Spezifikation ausschließlich mit GET dokumentiert. Für diese Bereiche ist kein POST-, PATCH- oder DELETE-Endpunkt dokumentiert.

- GET-only: abewegungen, adressarten, afa-saetze, anlagegruppen, anlagegueter, anschriftsrelationen, antragssperren, artikel, artikelbestaende, artikeleinheiten
- GET-only: as-bde, as-m-bewegung, as-m-stamm, as-s-bewegung, as-s-stamm, auszuege, banken, banktypen, bausteine, berichte
- GET-only: berufsbezeichnungen, berufskennzeichen, bestellanforderungen, bestellungen, beurteilungen, bezugsgroessen, bildkatalog, bildkatalog-link, bpositionen, buchungen
- GET-only: bundeslaender, choicelist, colorpalette, cronjobs, custom-fields, customfields, dashboards, datanorm-warengruppen, dbwidget, defaults

- GET-only: dhl-listen, dhl-productkeys, diktatzeichen, dimensionen, dimensionsnamen, dispoliste, eigene-konten, eingangsrechnungen, eingrechnungen, emailkonten
- GET-only: erechnungen, etiketten, etiketten-druck, eval, feature-flags, finanzamtmeldungen, firmengruppen, firmennetzwerke, firmenumsatz, flows
- GET-only: formulare, formularezv, freifeldcontainer, gefahrtarife, gruppen, gruppenitems, gui-views, hierarchien, hm-report, hmberichte
- GET-only: internetdaten, inventuren, inventuren-posten, itscope-hersteller, itscope-lieferanten, jobplanung, jobqueue, jobs, kalenderanhaenge, kalenderdaten
- GET-only: kalendereintragsarten, kasse-transaktionen, kassenbons, kategorien, kategorienamen, kbucheintraege, klr-salden, klr-texte, kombisuchen, kommunikationsdaten
- GET-only: kommunikationsserver, kontakte, konten, kontengruppen, kontrakte, kontrakte-posten, kostenstamm, krankenkassen, krankenkassenumlagen, kstgruppen
- GET-only: kstsalden-tage, kurse, laender, lager, leihsets, leihsets-list, lieferscheine, lizenzen, lohn-defaults, lohnabrechnungen
- GET-only: lohnarten, lohngruppen, mahntexte, mahnungen, mailings, mailings-einschraenkung, materialbelege, materialstapel, merkmale, merkmale-sprache
- GET-only: merkmalwerte, mietinformationen, mietinfos-relationen, mitarbeiter, mitarbeiterinfo, mitarbeitermodelle, mitarbeitersalden, mitarbeiterzugriffe, mitteilungen, mixed
- GET-only: modelle, modellitems, monatsabrechnung, monatsabrechnungen, monatszuschlaege, nachrichten, notizen, oe, pakete, pakete-packdaten
- GET-only: paletten, pflichtfelder, positionen, postleitzahlen, pr-job-fremd, pref-b, pref-g, pref-m, preise, preislisten
- GET-only: preisstrukturen, preisvorerfassung, prioritaeten, projektabos, projektabrechnung, projektbriefings, projekte, projektfavoriten, projekthierarchie, projektkonfig
- GET-only: projektmonatsabrechnung, projekttagesabrechnung, protokoll, prozesse, publikationen, rechnungen, reiseberechnung, reisekosten, reisekosten-anhang, reisekostenquittungen
- GET-only: reisekostenziele, reisetage, relarea, relate-bild-pubtext, relate-text-publikation, relate-text-style, reparaturen, ressourcen, ro-sntab, ro-status
- GET-only: rollen, rs-metadaten, schaeden, schichtmodelle, schichttage, schichtwoche, schichtwochenmodelle, serie, serviceobjekte, serviceprojekte
- GET-only: sets, so-relations, sozialmeldungen, standorte, stapelbelege, statushistorie, statuslisten, statuswerte, stilvorlagen, strukturen
- GET-only: stundenzettel, subetats, subetats-anleihen, subetats-verteilung, tagesabrechnungen, tagesmodelle, tageszuschlaege, tarifgruppen, tasks, tasks-arten

- GET-only: taxonomien, teams, teamzuordnungen, teleanrufe, teleindex, termineinstellungen, testlogin, texte, tickets, todos
- GET-only: touren, touren-positionen, transporte, umsaetze, unfallversicherungen, unitundabteilungen, ups-pickup, urlaubssalden, userexits, ustcodes
- GET-only: utest, verkaufschancen, waehrung-mittelkurse, waehrungen, wareneingaenge, wartungsarten-bewegung, wartungsarten-stamm, wartungsgruppen, wartungstermine, wochenabrechnung
- GET-only: wochenmodelle, wv, zaehler, zahlungen, zahlungsverlauf, zeitbereichszuschlaege, zeituebergabeanlohn, zeitzuschlaege, zubehoere