

Pflichtenheft – Company & Location Service

Service-Name: company-service
Version: V1.3
Stand: 21.02.2026
Ziel: Verwaltung von Unternehmens-Metadaten (Company) und Standorten (Location) in einer mandantenfähigen Plattform. V1.3 ergänzt Jurisdiktion (country/region/timezone) auf Location-Ebene und definiert verbindliche Docking-Contracts zu Contact- und Communication-Service.

1. Zweck und Zielsetzung (MUSS)

Der company-service MUSS:

- Companies als Metadaten-Objekte verwalten (Name, Logo, Defaults)
- Standorte (Locations) je Company verwalten
- Einen Hauptsitz (Headquarter) abbilden (über Location-Konzept)
- Für HR-/Zeiterfassungs-Use-Cases ausreichend Standortkontext liefern, ohne Adressen im Service zu speichern:
 - Zeitzone (IANA)
 - Land/Region (für Feiertage/Arbeitsregeln)

Der Service SOLL generisch bleiben, sodass er auch außerhalb HR nutzbar ist.

2. Geltungsbereich und Abgrenzung

2.1 In Scope (MUSS)

- CRUD für Company und Location (inkl. Soft-Delete/Restore sofern vorhanden)
- Standortattribute:
 - timezone (IANA)
 - locale (optional)
 - **V1.3 neu:** countryCode, regionCode (optional)
- Query-APIs für Location-Liste und Location-Details

2.2 Out of Scope (MUSS)

- Speicherung von postalischen Adressen, E-Mail, Telefon, Ansprechpartnern
- Versand/Benachrichtigung
- Arbeitsverträge, Rollen, Abwesenheiten, Zeiterfassung
- Feiertagsberechnung (nur Standortdaten liefern; Berechnung über Atlas/Holiday-Service)

3. Verantwortlichkeiten und Source of Truth (MUSS)

3.1 company-service ist Source of Truth für:

- Company Metadaten
- Location (organisatorischer Standort) inkl. Jurisdiktion: timezone, countryCode, regionCode

3.2 contact-service ist Source of Truth für:

- Adressen (postalisch), ContactPoints (E-Mail, Telefon etc.)
- Mehrfachadressen/Wohnorte
- Owner-Modell: ownerType + ownerId

3.3 communication-service ist Source of Truth für:

- Versandaufträge/Delivery-Status/Retry
- Empfängerauflösung über contact-service (nicht eigene Adresshaltung)

4. Begriffe (MUSS)

- **Company:** Metadaten eines Unternehmens/Mandanten (Name, Logo, Default-Einstellungen)
- **Location:** Standort/Arbeitsort-Kontext einer Company (organisatorisch), kann HQ sein
- **Jurisdiktion (Work Jurisdiction):** Kombination aus countryCode, regionCode, timezone für Feiertage/Arbeitsregeln
- **OwnerKey:** Schlüssel zum Andocken von Kontaktdaten: ownerType + ownerId

5. Jurisdiktion und Arbeitsortprinzip (V1.3)

5.1 Motivation

Feiertage/Arbeitsregeln hängen vom **Ort der Arbeitsleistung** ab. Firmenhauptsitz und Wohnort können davon abweichen.

5.2 Prinzip (MUSS)

Für Berechnungen (Feiertage/Regeln) gilt:

- maßgeblich ist der **tatsächliche Arbeitsort** bzw. die **Work Jurisdiction** des Einsatzes
- company-service liefert dafür Jurisdiktionsdaten an Locations

5.3 External Worksites (SOLL, systemweit)

Externe Einsatzorte (Kundenbaustellen) sind nicht zwingend Company-Locations.

Das System SOLL deshalb in shift/time-tracking später einen

`workJurisdictionOverride` unterstützen.

company-service bleibt hierbei unverändert (nicht zuständig für Kundenstandorte).

6. Datenmodell (V1.3)

6.1 CompanyEntity (MUSS)

Beispiel-Felder:

- `id` (String UUID/ULID)
- `name`
- `logoRef` (optional)
- Default `timezone / locale` (optional)
- `headquarterLocationId` (oder HQ-Flag an Location; je nach Implementierung)

6.2 LocationEntity (MUSS)

- `id` (String UUID/ULID)
- `companyId` (String)
- `name`
- `locationCode` (optional, falls vorhanden)
- `timezone` (String, IANA, optional; fallback via Company)
- **V1.3 neu:**
 - `countryCode` (String, ISO-2, optional)
 - `regionCode` (String, optional, empfohlen ISO-3166-2)

7. Validierung und Normalisierung (V1.3)

7.1 Normalisierung (MUSS)

- countryCode: trim, uppercase, "" → null
- regionCode: trim, uppercase, "" → null

7.2 Validierung (MUSS, wenn gesetzt)

- countryCode muss ^[A-Z]{2}\$ erfüllen
- regionCode max. 32 Zeichen (Pattern optional)

8. Docking Contract zu Contact-/Communication-Service (MUSS)

8.1 Owner Keys

Company

- ownerType = COMPANY
- ownerId = companyId

Location

- ownerType = LOCATION
- ownerId = locationId

8.2 Integrationsablauf (MUSS)

1. Company/Location wird in company-service erstellt
2. Address/ContactPoints werden im contact-service erstellt und referenzieren OwnerKey
3. communication-service versendet an Empfänger und löst ContactPoints über contact-service auf

8.3 Response Support (SOLL)

company-service SOLL in Responses die OwnerKeys zurückgeben:

- contactOwnerType, contactOwnerId
damit UI/Orchestrator keine OwnerKey-Logik duplizieren muss.

9. API (V1.3) – Überblick (MUSS)

Pfade exemplarisch; tatsächliche Base-Paths bleiben wie im Projekt.

Company:

- GET /api/v1/companies
- POST /api/v1/companies
- GET /api/v1/companies/{companyId}
- PATCH /api/v1/companies/{companyId}

Location:

- GET /api/v1/companies/{companyId}/locations
- GET /api/v1/locations/{locationId}
- PATCH /api/v1/locations/{locationId} (enthält V1.3 Felder)

10. Fehlerbehandlung (MUSS)

- 400 bei Validierungsfehlern (z. B. invalid countryCode)
- 404 bei nicht gefundenen Objekten
- 409 bei Version-Konflikt (falls Optimistic Lock)
- 401/403 bei Security/Tenant mismatch (je nach Sicherheitskonzept)

ErrorCodes (Beispiele):

- INVALID_COUNTRY_CODE
- INVALID_REGION_CODE
- LOCATION_NOT_FOUND
- TENANT_MISMATCH

11. Tests und Abnahmekriterien (V1.3)

11.1 Tests (MUSS)

- Positiv:
 - Update Location mit countryCode="de" → gespeichert/returned als DE
 - Update Location mit regionCode="de-hb" → DE-HB
- Negativ:
 - countryCode="D" oder "DEU" → 400

- Positiv:
 - CompanyResponse **enthält** contactOwnerType=COMPANY, contactOwnerId=companyId
 - LocationResponse **enthält** contactOwnerType=LOCATION, contactOwnerId=locationId

11.2 Abnahme (MUSS)

- Flyway Migration läuft auf leerer DB
- CRUD bleibt rückwärtskompatibel (neue Felder optional)
- Neue Felder in Swagger dokumentiert (Beispiele)
- mvn clean test grün

12. Nicht-Ziele (V1.3)

- Keine Adressen/Kontaktpunkte im company-service speichern
- Keine Feiertagsberechnung im company-service
- Keine HR-Regelengine im company-service