



Bundesverband für Energie- und
Wasserdatenmanagement e.V.

ERP-Implementierungsleitfaden

Billing Unit Data API v1.0

Version 1.0.3

Oktober 2026 | Zielgruppe: ERP-Anbieter / Softwareentwickler

ERP-Implementierungsleitfaden – bved Billing Unit Data API v1.0

Version: 1.0.3

Datum: 2026-10-01

Hinweis: Dieser Leitfaden wurde mit Hilfe von KI erzeugt.

Zielgruppe: ERP-Anbieter / Softwareentwickler (Wohnungswirtschaft)

Spezifikation: bved Billing Unit Data API v1.0.3 (OpenAPI: `bved-billing-unit-data-1-0.openapi.yaml`)

Inhalt

1. Einführung und Kontext
 2. Voraussetzungen
 3. Authentifizierung
 4. API-Endpunkte
 5. Vollständige Feldreferenz
 6. ERP-Datenmapping
 7. Implementierungsmuster
 8. Vollständige JSON-Beispiele
 9. Fehlerbehandlung
 10. Logging-Empfehlungen
 11. Implementierungs-Checkliste
 12. Testempfehlungen
-

1. Einführung und Kontext

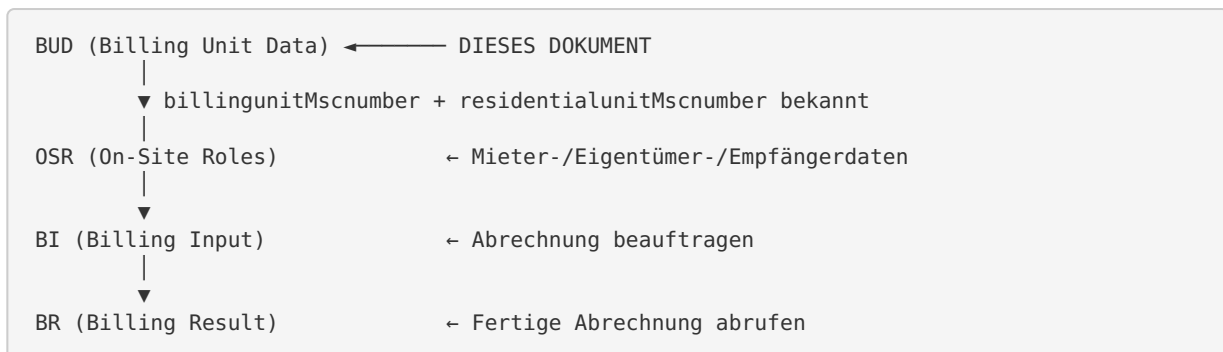
Was ist die Billing Unit Data API?

Die **bved Billing Unit Data API v1.0** (BUD) ist die REST-Schnittstelle, über die eine Hausverwaltung (Property Manager, PM) und ein Messdienstleister (Measuring Service Company, MSC) die **Stammdaten einer Abrechnungseinheit (AE)** und ihrer Nutzereinheiten (NE) abgleichen. Der Abgleich erfolgt nicht laufend, sondern **anlassbezogen** – bei neu vertraglich aufgenommenen Liegenschaften sowie bei strukturellen Änderungen (NE hinzugekommen, entfernt, geteilt, zusammengelegt). Den Abgleich **führt immer das PM-System (ERP) durch**, indem es ein Setup sendet. Den Bedarf an einem (erneuten) Abgleich erkennt das ERP am **Setup-Status der AE**: Eine AE steht auf **OPEN**, wenn sie neu vertraglich aufgenommen wurde oder der MSC sie für einen erneuten Abgleich geöffnet hat.

Übertragen werden:

- **Liste der Abrechnungseinheiten** beim MSC (Lesen) – inklusive MSC-Nummer, PM-Nummer, Adresse und Setup-Status
- **Setup-Daten einer Abrechnungseinheit** (Schreiben) – Wohneinheiten mit Lage-, Flächen- und letzten bekannten Empfängerdaten zum Matching beim MSC
- **Setup-Ergebnis** (Lesen) – vom MSC vergebene Ordnungsbegriffe (MSC-Nummern) der Nutzeinheiten und MSC-seitige Zusatz-Einheiten

Einordnung in die bved-API-Familie



Position im Prozess: BUD ist die **erste API**, die ein ERP nach Vertragsschluss mit einem MSC bedient. Erst wenn die AE beim MSC angelegt und die Nutzeinheiten gemappt sind, können OSR, BI und BR genutzt werden.

Kernprinzip: Setup-Workflow in drei Schritten

1. GET /billingunits → Welche AE kennt der MSC bereits? Welche brauchen Setup?
2. POST /billingunits/setup/{...} → PM sendet seine vorhandenen Informationen zu seinen Nutzeinheiten an den MSC
3. GET /billingunits/setup/{...} → MSC liefert MSC-Nummern der gematchten NE zurück

Der vertragliche Rahmen (welche AE der PM beim MSC führt) wird **außerhalb der API** vereinbart. Die API arbeitet ausschließlich auf bereits vertraglich vereinbarten AE.

Setup-Status der Abrechnungseinheit

Jede AE hat beim MSC einen Setup-Status, der den Bearbeitungsstand des vom PM gesendeten Setups widerspiegelt. Der Status ist die zentrale Steuerinformation des BUD-Workflows: Er zeigt, ob für eine AE noch ein Erst-Setup aussteht, ob ein gesendetes Setup beim MSC bearbeitet wird oder ob das Ergebnis abrufbar ist.

Status	Bedeutung	ERP-Maßnahme
OPEN	Ein Abgleich steht aus – entweder eine neu vertraglich aufgenommene AE ohne Erst-Setup oder eine bereits gemappte AE, die der MSC für einen erneuten Abgleich geöffnet hat	POST /setup mit aktuellem Stand ausführen
IN_PROGRESS	Ein Setup wurde vom PM gesendet und wird beim MSC bearbeitet	Auf COMPLETED warten, kein neues POST zulässig
COMPLETED	Setup ist beim MSC abgeschlossen, MSC-Nummern der NE sind verfügbar	GET /setup aufrufen, MSC-Nummern und ggf. additional[] ins ERP übernehmen

Wer löst was aus: Den Abgleich – also das Senden des Setups – führt **immer der PM** durch. Den Status **OPEN** kann jedoch **auch der MSC** setzen: Er kann eine bereits gemappte AE ohne vorheriges PM-POST wieder auf **OPEN** setzen, wenn sich die Struktur beim MSC geändert hat oder aus Sicht des MSC ein erneuter Abgleich erforderlich ist. Das ERP muss den Status daher **regelmäßig** prüfen – nicht nur unmittelbar nach eigenen POST-Vorgängen – und auf eine vom MSC geöffnete AE mit einem erneuten Setup reagieren.

2. Voraussetzungen

Voraussetzung	Details
OAuth 2.0 Credentials	Client ID und Client Secret vom MSC – nötig für Token-Abruf
Vertraglich vereinbarte AE	Welche AE der PM beim MSC führt, wird außerhalb der API geklärt. Die AE muss beim MSC angelegt sein, sonst gibt es kein <code>billingunitMscnumber</code>
<code>billingunitMscnumber</code>	9-stellige MSC-Nummer der AE – wird vom MSC beim vertraglichen Onboarding mitgeteilt und ist in <code>GET /billingunits</code> enthalten
PM-seitige Stammdaten	Adressen, Wohneinheiten mit Flächen, Lage, Etage, aktuelle Hauptmieter zum Matching
HTTPS/TLS	Alle API-Aufrufe ausschließlich über verschlüsselte Verbindung
Basis-URL	Wird vom jeweiligen MSC bereitgestellt – kein gemeinsamer Endpunkt, jeder MSC hat seine eigene URL

3. Authentifizierung

Die API ist mit **OAuth 2.0 Client Credentials Flow** gesichert.

Token-Abruf

```
POST {tokenUrl}
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id={client_id}
&client_secret={client_secret}
```

Die `tokenUrl` wird vom MSC mitgeteilt.

Token-Verwendung

```
GET /billingunitdata/v1/billingunits
Authorization: Bearer {access_token}
```

Implementierungshinweise

- Access Token vor Ablauf erneuern (TTL ist MSC-spezifisch — typisch 3600 Sekunden)
- Token serverseitig cachen, nicht per Request neu abrufen
- Bei `401`-Antwort: Token erneuern und Request einmalig wiederholen
- Credentials nicht im Quellcode oder in Logs speichern
- Der gleiche Token kann für alle bved-APIs desselben MSC wiederverwendet werden, sofern der MSC das so vorsieht

4. API-Endpunkte

Basis-URL: `{msc-base-url}/billingunitdata/v1`

Alle Antworten enthalten den Response-Header `X-API-VERSION` mit der vom MSC implementierten Version.

4.1 GET — Liste aller Abrechnungseinheiten

```
GET /billingunits
GET /billingunits?from=YYYY-MM-DD
GET /billingunits?status=OPEN|IN_PROGRESS|COMPLETED
GET /billingunits?offset=0&limit=50
```

Liefert eine paginierte Liste aller dem aktuellen Benutzer beim MSC zugeordneten AE — inklusive `billingunitMscnumber`, `setupstatus` und Adresse.

Query-Parameter:

Parameter	Typ	Pflicht	Beschreibung
<code>from</code>	date	Nein	Nur AE, deren <code>lastupdate</code> $\geq$ <code>from</code> ist
<code>status</code>	enum	Nein	Filter auf einen Setup-Status (<code>OPEN</code> , <code>IN_PROGRESS</code> , <code>COMPLETED</code>)
<code>offset</code>	int	Nein	Anzahl zu überspringender Einträge (Default 0)
<code>limit</code>	int	Nein	Max. Einträge pro Antwort, Vielfaches von 10, max. 100 (Default 20)

Antwort bei Erfolg: **200 OK**

```
{
  "totalentries": 42,
  "billingunits": [
    {
      "billingunitMscnumber": "001670082",
      "billingunitPmnumber": "B120586",
      "customerMscnumber": "4000000001",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "lastupdate": "2025-01-01T12:13:07Z",
      "setupstatus": "IN_PROGRESS"
    }
  ]
}
```

Verwendungszweck: Erstabruf aller Abrechnungseinheiten; regelmäßiger Status-Check, um neu vom MSC freigeschaltete (vertraglich aufgenommene) AE sowie vom MSC erneut auf `OPEN` gesetzte (bereits gemappte) AE zu erkennen und um nach eigenem `POST /setup` den Übergang von `IN_PROGRESS` auf `COMPLETED` zu verfolgen.

4.2 POST – Setup einer Abrechnungseinheit senden

```
POST /billingunits/setup/{billingunit}
Content-Type: application/json
```

Sendet die vom PM bekannten Wohneinheiten zur Abrechnungseinheit, damit der MSC sie mit seinen eigenen Einheiten matchen kann.

Antwort bei Erfolg: **200 OK**

```
{
  "transactionid": "T-0000042"
}
```

Die `transactionid` identifiziert den Vorgang eindeutig beim MSC und ist für Support-Anfragen essenziell. Nach erfolgreichem POST wechselt der Setup-Status der AE auf `IN_PROGRESS`.

Antwort bei Validierungsfehlern: 400 Bad Request

```
{
  "transactionid": "T-0000099",
  "messages": [
    {
      "type": "ERROR",
      "shortmessage": "INCOMPLETE",
      "message": "residentialunits[2].positiontext fehlt",
      "language": "de"
    }
  ]
}
```

4.3 GET — Setup-Ergebnis einer Abrechnungseinheit

```
GET /billingunits/setup/{billingunit}
```

Liefert das Ergebnis des Matchings beim MSC, sobald der Setup-Status der AE **COMPLETED** ist. Solange die AE noch auf **OPEN** oder **IN_PROGRESS** steht, existiert **kein Ergebnis** — der MSC antwortet dann mit **404** (kein Ergebnis vorhanden, siehe §9.1). Das ERP ruft diesen Endpunkt daher erst auf, nachdem die AE über den Status-Check (§7.4) auf **COMPLETED** gewechselt ist.

Antwort bei Erfolg: 200 OK

```
{
  "billingunitMscnumber": "001670082",
  "billingunitPmnumber": "B120586",
  "matched": [
    {
      "residentialunitMscnumber": "0001",
      "residentialunitPmnumber": "B120586-001",
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-001/01"
      }
    }
  ],
  "additional": [
    {
      "residentialunitMscnumber": "0007",
      "positiontext": "DG re",
      "address": { "street": "Möhrenstr. 56", "postalcode": "05555", "city": "Musterhausen", "country": "DEU" }
    }
  ]
}
```

matched[] — Einheiten, die vom PM gesendet **und** vom MSC erkannt wurden. Jeder Eintrag enthält die jetzt verbindlich gültige **residentialunitMscnumber** — diese muss im ERP gespeichert werden, da sie in OSR, BI, BR und CD (uVI) als Path-Parameter benötigt wird.

additional[] — Einheiten, die der MSC selbst kennt, die der PM aber nicht gesendet hat. Mögliche Ursachen: Datenpflegelücke beim PM, abweichende Strukturierung (Teilung/Zusammenlegung). Das ERP muss diese Einträge dem Sachbearbeiter zur Klärung anzeigen. Solange

für eine solche Einheit keine PM-Referenz existiert, kann das ERP für sie **kein OSR senden und sie nicht über BI aktualisieren** — sie sind aber dennoch Teil der Abrechnung und werden ggf. im Billing Result (BR) vom MSC übermittelt. Für die gematchten Einheiten (`matched[]`) ist OSR/BI dagegen uneingeschränkt möglich.

4.4 Path-Parameter

Parameter	Feld im Body	Typ	Länge	Beispiel
<code>billingunit</code>	<code>billingunitMscnumber</code>	string	9 Stellen	<code>001670082</code>

Path-Parameter und das `billingunitMscnumber` -Feld im Body müssen konsistent sein.

5. Vollständige Feldreferenz

5.1 BillingUnit (Antwort von `GET /billingunits`)

Feld	Typ	Pflicht	Länge / Format	Beschreibung
<code>billingunitMscnumber</code>	string	Ja	9 Stellen	MSC-Nummer der AE
<code>billingunitPmnumber</code>	string	Nein	1–15 Zeichen	ERP-eigene Nummer der AE
<code>customerMscnumber</code>	string	Ja	1–10 Zeichen	Kundennummer des PM beim MSC
<code>address</code>	Address	Nein	—	Anschrift der Liegenschaft
<code>lastupdate</code>	date-time	Ja	ISO 8601	Zeitpunkt der letzten Aktualisierung beim MSC
<code>setupstatus</code>	enum	Ja	OPEN / IN_PROGRESS / COMPLETED	Setup-Status der AE

5.2 SetupRequest (POST Request Body)

Feld	Typ	Pflicht	Länge / Format	Beschreibung
<code>billingunitMscnumber</code>	string	Ja	9 Stellen	MSC-Nummer der AE — muss mit Path-Parameter übereinstimmen

Feld	Typ	Pflicht	Länge / Format	Beschreibung
billingunitPmnumber	string	Nein	1–15 Zeichen	ERP-eigene Nummer der AE (z.B. interne Liegenschaftsnummer)
customerMscnumber	string	Ja	1–10 Zeichen	Kundennummer des PM beim MSC
description	string	Nein	1–35 Zeichen	Unstrukturierte Zusatzinfo zur Liegenschaft (Adresszeile o.ä.) – wird vom MSC ggf. nur als Matching-Hilfe verwendet und nicht gespeichert
residentialunits	array	Ja	min. 1 Eintrag	Liste der Nutzeinheiten der AE

5.3 SetupRequestResidentialUnit

Feld	Typ	Pflicht	Länge / Format	Beschreibung
residentialunitMscnumber	string	Nein	4 Stellen	MSC-Nummer der NE, falls vom MSC bereits bekannt – andernfalls leer lassen
residentialunitPmnumber	string	Nein	1–20 Zeichen	ERP-eigene Nummer der NE (z.B. B120586-001)
address	Address	Nein	–	Anschrift der NE (oft identisch mit AE-Adresse)
positiontext	string	Nein	1–20 Zeichen	Lagebezeichnung wie auf Abrechnung gedruckt (z.B. 1 OG re)
validfrom	date	Nein	YYYY-MM-DD	Gültig-ab-Datum, wenn nicht sofort gültig
floornumber	integer	Nein	1–999	Etagennummer
floortype	enum	Nein	siehe unten	Etagentyp
positiondetail	string	Nein	1–6 Zeichen	Zusatz zur Lage (z.B. re , li , Mitte)
entrancenumber	string	Nein	1–10 Zeichen	Aufgang / in Österreich „Stiege“
doornumber	string	Nein	1–10 Zeichen	Türnummer

Feld	Typ	Pflicht	Länge / Format	Beschreibung
<code>description</code>	string	Nein	1–35 Zeichen	Unstrukturierte Zusatzinfo (z.B. „Ladenlokal“) – wird vom MSC nicht zwingend gespeichert
<code>heatedfloorspace</code>	number	Nein	0.0–99999999.99	Beheizte Wohnfläche (m ²)
<code>totalfloorspace</code>	number	Nein	0.0–99999999.99	Gesamtwohnfläche (m ²)
<code>heatedenclosedspace</code>	number	Nein	0.0–99999999.99	Beheizter umbauter Raum (m ³)
<code>totalenclosedspace</code>	number	Nein	0.0–99999999.99	Gesamter umbauter Raum (m ³)
<code>currentbillingrecipient</code>	object	Nein	–	Aktueller Hauptmieter – nur zum Matching

floortype – gültige Werte:

Wert	Bedeutung
<code>UPPER_FLOOR</code>	Obergeschoss
<code>GROUND_FLOOR</code>	Erdgeschoss
<code>BASEMENT</code>	Untergeschoss
<code>ATTIC</code>	Dachgeschoss
<code>MEZZANINE</code>	Zwischengeschoss
<code>CELLAR</code>	Keller

currentbillingrecipient – CurrentBillingRecipient-Objekt:

Feld	Typ	Pflicht	Länge	Beschreibung
<code>partnerPmnumber</code>	string	Nein	1–20 Zeichen	ERP-ID des aktuellen Hauptmieters
<code>name</code>	string	Nein	1–35 Zeichen	Nachname (Privatperson) oder Firmenname
<code>firstname</code>	string	Nein	1–35 Zeichen	Vorname (nur natürliche Personen)

Feld	Typ	Pflicht	Länge	Beschreibung
<code>additionalname</code>	string	Nein	1–35 Zeichen	Namenszusatz (nicht alle MSC unterstützen dieses Feld)

Der `currentbillingrecipient` im Setup dient **ausschließlich dem Matching** der NE beim MSC. Die endgültigen Empfängerdaten werden später über die OSR-API übertragen und sind dort verbindlich.

`address` – Address-Objekt:

Feld	Typ	Pflicht	Länge	Beschreibung
<code>street</code>	string	Nein	1–35 Zeichen	Straße und Hausnummer
<code>postalcode</code>	string	Nein	1–10 Zeichen	Postleitzahl
<code>city</code>	string	Nein	1–35 Zeichen	Stadt
<code>country</code>	string	Nein	genau 3 Zeichen	ISO 3166-1 alpha-3 (z.B. DEU , AUT , CHE)

5.4 SetupResult (Antwort von `GET /billingunits/setup/{...}`)

Feld	Typ	Pflicht	Beschreibung
<code>billingunitMscnumber</code>	string	Ja	MSC-Nummer der AE
<code>billingunitPmnumber</code>	string	Nein	ERP-eigene Nummer der AE (vom MSC zurückgespiegelt)
<code>matched</code>	array	Ja	NE, die vom PM gesendet und vom MSC erkannt wurden – mit jetzt gültiger <code>residentialunitMscnumber</code>
<code>additional</code>	array	Nein	NE, die der MSC kennt, die der PM aber nicht gesendet hat

`matched[]` – SetupResultResidentialUnit:

Feld	Typ	Pflicht	Beschreibung
<code>residentialunitMscnumber</code>	string	Ja	Verbindliche MSC-Nummer der NE – in ERP übernehmen
<code>residentialunitPmnumber</code>	string	Nein	Vom PM ursprünglich gesendete PM-Nummer (Korrelation)
<code>currentbillingrecipient</code>	object	Nein	Vom PM gesendeter Hauptmieter (zur Korrelation)

additional[] — SetupResultAdditional:

Feld	Typ	Pflicht	Beschreibung
<code>residentialunitMscnumber</code>	string	Ja	MSC-Nummer der zusätzlichen NE
<code>positiontext</code>	string	Ja	Lagebezeichnung, wie der MSC sie kennt
<code>currentbillingrecipient</code>	object	Nein	Aktueller Empfänger laut MSC
<code>address</code>	Address	Nein	Anschrift laut MSC

6. ERP-Datenmapping

6.1 ERP-Liegenschaft → `SetupRequest`

ERP-Feld	API-Feld	Hinweis
Liegenschafts-/ Objektnummer	<code>billingunitPmnumber</code>	Optionale Referenz für Wiedererkennung beim MSC
Kundennummer beim MSC	<code>customerMscnumber</code>	Aus vertraglichem Onboarding bekannt
Adresszeile / Bezeichner	<code>description</code>	Frei, dient nur dem Matching
MSC-AE-Nummer	<code>billingunitMscnumber</code>	Aus <code>GET /billingunits</code> übernommen

6.2 ERP-Wohneinheit → `SetupRequestResidentialUnit`

ERP-Feld	API-Feld	Besonderheit
Wohnungsnummer (intern)	<code>residentialunitPmnumber</code>	Stabiler ERP-Schlüssel — niemals beim Mieterwechsel ändern
Anschrift NE	<code>address</code>	Bei klassischen Mehrfamilienhäusern oft identisch mit AE-Adresse
Lagebezeichnung (Druck)	<code>positiontext</code>	Z.B. <code>EG re</code> , <code>1. OG li</code> , <code>DG</code>
Etage (numerisch)	<code>floornumber</code>	1 = Erdgeschoss/erste Etage, je nach ERP-Konvention
Etagentyp	<code>floortype</code>	ERP-Etagencodes auf Enum mappen
Lage-Detail	<code>positiondetail</code>	Kurzcode wie <code>re</code> , <code>li</code> , <code>Mitte</code> , <code>vorn</code> , <code>hinten</code>

ERP-Feld	API-Feld	Besonderheit
Aufgang	<code>entrancenumber</code>	In Deutschland Hausaufgang, in Österreich „Stiege“
Türnummer	<code>doornumber</code>	Falls am Türschild vorhanden
Wohnfläche beheizt (m ²)	<code>heatedfloorspace</code>	Dezimaltrennzeichen ist Punkt
Wohnfläche gesamt (m ²)	<code>totalfloorspace</code>	
Umbauter Raum beheizt (m ³)	<code>heatedenclosedspace</code>	Nur bei Bedarf, oft nicht gepflegt
Umbauter Raum gesamt (m ³)	<code>totalenclosedspace</code>	
Gültig ab	<code>validfrom</code>	Nur bei zukünftig entstehenden NE (Neubau, Umbau, Teilung) befüllen
Aktueller Hauptmieter	<code>currentbillingrecipient</code>	Identifizierungshilfe für MSC — keine fachliche Empfängerinformation

6.3 Was `currentbillingrecipient` ist (und was es nicht ist)

Ist	Ist nicht
Identifizierungshilfe für Matching beim MSC	Verbindliche Empfängerangabe für Abrechnung
Snapshot des aktuellen Hauptmieters zum Zeitpunkt des Setups	Vertragsdatensatz (kein <code>validfrom</code> / <code>validto</code>)
Nützlich zur Disambiguierung gleicher Adressen	Ersatz für die OSR-API

Verbindliche Empfängerdaten gehören in die OSR-API. Der `currentbillingrecipient` im BUD-Setup wird vom MSC ausschließlich zum Matching genutzt, nicht zur Abrechnung.

6.4 Stabilität der `residentialunitPmnumber`

Die `residentialunitPmnumber` ist **optional** (siehe §5.3). Wird sie verwendet — was dringend empfohlen ist —, ist sie der **stabile Korrelations-Schlüssel** zwischen PM- und MSC-System und muss dann:

- Beim Mieterwechsel unverändert bleiben (an die Wohnung, nicht an den Mieter gebunden)
- In allen Folge-APIs (OSR, BI, BR, CD) unverändert weiterverwendet werden

- Eindeutig innerhalb einer AE sein

Wenn ein ERP eine einmal vergebene `residentialunitPmnumber` bei Mieterwechsel ändert, geht die Korrelation verloren – der MSC erkennt die Wohnung im nächsten Setup nicht wieder, und das Matching schlägt fehl.

Wird **keine** `residentialunitPmnumber` gesendet, muss der MSC die NE allein über die übrigen Matching-Attribute (Adresse, `positiontext`, Lage, `currentbillingrecipient`) zuordnen – das ist fehleranfälliger. Eine stabile PM-Nummer wird daher dringend empfohlen.

6.5 Umgang mit `additional[]` aus dem Setup-Ergebnis

Wenn der MSC NE meldet, die das ERP nicht kennt, kann das bedeuten:

Ursache	ERP-Reaktion
NE im ERP fehlt (Datenlücke)	Sachbearbeiter anlegen, beim nächsten Setup mitsenden
NE wurde im ERP geteilt/zusammengelegt, aber nicht beim MSC	Klärung mit MSC; ggf. erneut Setup senden
NE existiert nur beim MSC (Altdaten)	MSC bitten, die NE zu deaktivieren

Die `additional[]`-Liste sollte im ERP **immer dem Sachbearbeiter angezeigt** werden, nie still verworfen. Solange NE nur in `additional[]` stehen (keine PM-Referenz), kann das ERP für sie kein OSR senden und sie nicht über BI aktualisieren. Im Billing Result (BR) können sie dennoch erscheinen, da sie beim MSC Teil der Abrechnung sind.

7. Implementierungsmuster

7.1 Grundregel: Vollständiges Setup je AE

Ein POST-Setup enthält die **vollständige Wohneinheits-Struktur** einer AE – alle aktiven NE, nicht nur die geänderten. Der MSC ersetzt sein bisheriges Bild der AE-Struktur durch das Gesendete und matched anschließend.

7.2 Wann senden?

BUD ist **anlassbezogen**, nicht laufend. Ein POST findet nur bei einem der folgenden Ereignisse statt:

Trigger-Ereignis	Empfehlung
Neue Liegenschaft beim MSC unter Vertrag (erscheint als <code>OPEN</code> in <code>GET /billingunits</code>)	Nach Erhalt der <code>billingunitMscnumber</code> Erst-Setup senden

Trigger-Ereignis	Empfehlung
AE vom MSC erneut auf OPEN gesetzt (Strukturänderung beim MSC / MSC-seitiger Abgleichbedarf)	Erneutes Setup mit aktuellem Stand senden
NE neu hinzugekommen (Anbau, Teilung)	Erneutes Setup mit vollständiger Struktur
NE entfernt (Zusammenlegung, Rückbau)	Erneutes Setup mit reduzierter Struktur
Adresse oder Flächendaten strukturell korrigiert	Erneutes Setup mit korrigierten Daten

Kein POST nötig bei: Mieterwechsel, Empfängeränderung, Verbrauchsabrechnung, EED-Versand. Diese Vorgänge laufen über OSR/BI/BR/CD und triggern keinen BUD-Abgleich.

Erneuter Abgleich: Ein erneutes Setup führt stets der PM durch. Auslöser kann eine Änderung im ERP sein **oder** eine vom MSC auf **OPEN** zurückgesetzte AE (Strukturänderung beim MSC bzw. MSC-seitiger Abgleichbedarf). Letztere erkennt das ERP nur über den regelmäßigen Status-Check (siehe 7.4).

7.3 Muster 1: Erstabruf aller Abrechnungseinheiten

Eine neue Liegenschaft wird beim MSC unter Vertrag genommen.

1. GET /billingunits?status=OPEN
→ liefert AE mit billingunitMscnumber = "001670082", setupstatus = OPEN
2. POST /billingunits/setup/001670082
Body: { billingunitMscnumber, customerMscnumber, residentialunits: [alle NE mit Pmnumber, Adresse, Lage, Fläche, currentbillingrecipient] }
→ 200 + transactionid
3. Polling: GET /billingunits?status=COMPLETED (oder zeitgesteuert)
→ AE erscheint mit setupstatus = COMPLETED
4. GET /billingunits/setup/001670082
→ matched[].residentialunitMscnumber ins ERP übernehmen, gemappt über matched[].residentialunitPmnumber

7.4 Muster 2: Regelmäßiger Status-Check

Der Status ändert sich durch das eigene **POST /setup** des PM (**OPEN** / **COMPLETED** → **IN_PROGRESS**) und durch den Abschluss der Bearbeitung beim MSC (**IN_PROGRESS** → **COMPLETED**). Zusätzlich kann der MSC den Status auf **OPEN** setzen: neu vertraglich aufgenommene AE erscheinen als **OPEN**, und eine bereits gemappte AE kann der MSC erneut auf **OPEN** setzen, wenn aus seiner Sicht ein neuer Abgleich nötig ist. Ein **regelmäßiger Status-Check** der gesamten AE-Liste ist daher erforderlich – unabhängig davon, ob der PM zuletzt selbst etwas gesendet hat.

Empfohlene Strategie (täglich, z.B. via Cron):

```
GET /billingunits?from={letzter-sync-tag}
→ für jede AE im Ergebnis:
    OPEN      → Setup mit aktuellem Stand senden (Erst-Setup neuer AE oder vom MSC
angeforderter erneuter Abgleich)
    IN_PROGRESS → nichts tun, wird beim nächsten Lauf erneut geprüft
    COMPLETED → GET /setup aufrufen, Ergebnis ins ERP übernehmen
```

Push-Mechanismen vom MSC gibt es nicht – die API kennt nur Polling.

Nach einem eigenen `POST /setup` wird der Status im Rahmen des täglichen Laufs weiter beobachtet, bis die AE auf `COMPLETED` wechselt und die `residentialunitMscnumber` ins ERP übernommen werden kann. Die Bearbeitung beim MSC kann **mehrere Tage** in Anspruch nehmen – sie erfolgt teilweise manuell (z.B. bei Datendifferenzen). Ein häufigeres Polling als einmal in 24 Stunden ist daher nicht sinnvoll. Solange die AE auf `IN_PROGRESS` steht, ist ein erneutes `POST /setup` nicht zulässig (409, siehe §9.1); der Vorgang darf im ERP nicht als Fehler behandelt werden, nur weil er über mehrere Tagesläufe hinweg offen bleibt.

7.5 Muster 3: Korrektur einer bestehenden AE

Beispiel: NE 0003 wird im ERP geteilt in 0003 und 0008.

1. ERP-Stand aktualisieren: zwei NE statt einer (`residentialunitPmnumber` beibehalten für 0003, neue `Pmnumber` für 0008).
2. `POST /billingunits/setup/001670082`
Body enthält alle bisherigen NE + die neue NE 0008.
→ status der AE wechselt auf `IN_PROGRESS`
3. `GET /setup` nach Abschluss → MSC vergibt `residentialunitMscnumber` für die neue NE.

Wichtig: Niemals nur die geänderten NE senden. Der MSC ersetzt das vorhandene Bild der AE-Struktur durch den gesendeten Stand.

7.6 Muster 4: Paginierung beim Inventar-Abruf

Bei großen PM-Mandaten (>100 AE):

```
offset = 0
do {
    GET /billingunits?offset={offset}&limit=100
    → speichere billingunits[]
    offset += 100
} while (offset < totalentries)
```

`totalentries` kommt als Pflichtfeld mit jeder Antwort – die erste Seite der Schleife liefert es bereits; ein separater Vorab-Aufruf ist nicht nötig.

7.7 Muster 5: Inkrementeller Abgleich mit `from`

Tägliche Synchronisation:

```
GET /billingunits?from={letzter-sync-tag}
```

→ liefert nur AE, deren lastupdate seit dem letzten Sync verändert wurde

Lokal letzten Sync-Zeitstempel persistieren. Nach erfolgreicher Verarbeitung aktualisieren.

7.8 Idempotenz nutzen

POST `/billingunits/setup/{...}` ist by design idempotent: Dasselbe Request erneut senden hat keine unerwünschten Nebeneffekte (der MSC ersetzt sein Bild durch den gleichen Inhalt). Dieses Verhalten ermöglicht sichere Retry-Logik:

1. Request-Body lokal puffern
2. POST senden
3. Netzwerkfehler / Timeout → identischen Body direkt erneut senden
4. Erst nach 200-Antwort als erfolgreich markieren
5. Keine kumulative Zustandsverwaltung nötig

8. Vollständige JSON-Beispiele

Grundregel: Jeder POST-Setup enthält den **vollständigen Strukturstand der AE** – alle NE, nicht nur geänderte. Der MSC ersetzt sein bisheriges Bild.

Alle Beispiele verwenden eine gemeinsame AE:

Feld	Wert
<code>billingunitMscnumber</code>	<code>001670082</code>
<code>billingunitPmnumber</code>	<code>B120586</code>
<code>customerMscnumber</code>	<code>4000000001</code>
Adresse	Möhrenstr. 56, 05555 Musterhausen, DEU

8.1 GET /billingunits – Liste der AE (Antwort)

```
{
  "totalentries": 1,
  "billingunits": [
    {
      "billingunitMscnumber": "001670082",
      "billingunitPmnumber": "B120586",
      "customerMscnumber": "4000000001",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "lastupdate": "2026-05-19T08:30:00Z",
      "setupstatus": "OPEN"
    }
  ]
}
```

8.2 POST /billingunits/setup/{billingunit} – Setup senden (Request)

Drei NE: Erdgeschoss, 1. OG rechts, 1. OG links. Aktuelle Hauptmieter werden zur Identifikation mitgeschickt.

```

{
  "billingunitMscnumber": "001670082",
  "billingunitPmnumber": "B120586",
  "customerMscnumber": "4000000001",
  "description": "Möhrenstr. 56",
  "residentialunits": [
    {
      "residentialunitPmnumber": "B120586-001",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "positiontext": "EG",
      "floornumber": 1,
      "floortype": "GROUND_FLOOR",
      "entrancenummer": "1",
      "doornumber": "33a",
      "heatedfloorspace": 70.74,
      "totalfloorspace": 77.81,
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-001/01",
        "name": "Holstein",
        "firstname": "Horst"
      }
    },
    {
      "residentialunitPmnumber": "B120586-002",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "positiontext": "1. OG re",
      "floornumber": 2,
      "floortype": "UPPER_FLOOR",
      "positiondetail": "re",
      "entrancenummer": "1",
      "heatedfloorspace": 54.22,
      "totalfloorspace": 59.64,
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-002/01",
        "name": "Schleswig",
        "firstname": "Frieda"
      }
    },
    {
      "residentialunitPmnumber": "B120586-003",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "positiontext": "1. OG li",
      "floornumber": 2,
      "floortype": "UPPER_FLOOR",
      "positiondetail": "li",
      "entrancenummer": "1",
      "heatedfloorspace": 32.20,
      "totalfloorspace": 35.42,
    }
  ]
}

```

```
"currentbillingrecipient": {  
  "partnerPmnumber": "B120586-003/01",  
  "name": "Lorenz",  
  "firstname": "Christel"  
}  
]  
}
```

8.3 POST /billingunits/setup/{billingunit} – Erfolgs-Antwort

```
{  
  "transactionid": "T-0000042"  
}
```

8.4 GET /billingunits/setup/{billingunit} – Setup-Ergebnis nach Abschluss

Der MSC hat alle drei NE gemappt und seine Ordnungsbegriffe (`residentialunitMscnumber`) vergeben. Es gibt zusätzlich eine MSC-seitige NE, die der PM nicht gesendet hat (z.B. ein bisher nicht gepflegtes Dachgeschoss).

```

{
  "billingunitMscnumber": "001670082",
  "billingunitPmnumber": "B120586",
  "matched": [
    {
      "residentialunitMscnumber": "0001",
      "residentialunitPmnumber": "B120586-001",
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-001/01",
        "name": "Holstein",
        "firstname": "Horst"
      }
    },
    {
      "residentialunitMscnumber": "0002",
      "residentialunitPmnumber": "B120586-002",
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-002/01",
        "name": "Schleswig",
        "firstname": "Frieda"
      }
    },
    {
      "residentialunitMscnumber": "0003",
      "residentialunitPmnumber": "B120586-003",
      "currentbillingrecipient": {
        "partnerPmnumber": "B120586-003/01",
        "name": "Lorenz",
        "firstname": "Christel"
      }
    }
  ],
  "additional": [
    {
      "residentialunitMscnumber": "0007",
      "positiontext": "DG re",
      "address": {
        "street": "Möhrenstr. 56",
        "postalcode": "05555",
        "city": "Musterhausen",
        "country": "DEU"
      },
      "currentbillingrecipient": {
        "name": "Berger"
      }
    }
  ]
}

```

Nach diesem Schritt sind die `residentialunitMscnumber` (0001, 0002, 0003) im ERP zu speichern und können in OSR/BI/BR als Path-Parameter verwendet werden. Die `additional` - Einheit 0007 muss dem Sachbearbeiter zur Klärung vorgelegt werden.

8.5 POST mit Validierungsfehler — Antwort

```
{
  "transactionid": "T-0000099",
  "messages": [
    {
      "type": "ERROR",
      "shortmessage": "INCOMPLETE",
      "message": "residentialunits[2].positiontext fehlt",
      "language": "de"
    },
    {
      "type": "WARNING",
      "shortmessage": "IMPLAUSIBLE",
      "message": "heatedfloorspace > totalfloorspace bei NE B120586-002",
      "language": "de"
    }
  ]
}
```

9. Fehlerbehandlung

9.1 HTTP-Statuscodes

Status	Bedeutung	ERP-Maßnahme
200	Daten erfolgreich übernommen / Liste erfolgreich abgerufen	<code>transactionid</code> protokollieren, ggf. Setup-Status weiter verfolgen
400	Validierungsfehler (fachlich oder technisch)	<code>ValidationResponse.messages</code> auslesen, Fehler dem Sachbearbeiter anzeigen
401	Token ungültig oder abgelaufen	Token neu abrufen, Request einmalig wiederholen
403	Keine Berechtigung	Vertragliche Klärung mit MSC; keine automatische Wiederholung
404	AE beim MSC unbekannt oder — bei <code>GET /setup/{billingunit}</code> — für die AE liegt (noch) kein Setup-Ergebnis vor (Setup nicht <code>COMPLETED</code>)	Bei <code>GET /setup</code> : Status-Check gemäß §7.4, erst bei <code>COMPLETED</code> erneut aufrufen. Bei unbekannter AE: <code>billingunitMscnumber</code> prüfen, vorher <code>GET /billingunits</code> aufrufen
409	Für diese AE wird bereits ein Setup vom MSC bearbeitet (AE-Status <code>IN_PROGRESS</code>) — ein paralleles <code>POST /setup</code> ist nicht zulässig	Kein erneutes POST; auf <code>COMPLETED</code> warten (Status-Check gemäß §7.4), erst dann ggf. neu senden. Antwort enthält eine <code>ValidationResponse</code>
500	Technischer Fehler beim MSC	Exponentiellen Backoff anwenden; MSC mit <code>transactionid</code> (falls vorhanden) kontaktieren

9.2 ValidationResponse – shortmessage Routing

shortmessage	Typische Ursache bei BUD	Lösungsansatz
INCOMPLETE	Pflichtfeld fehlt (z.B. <code>customerMscnumber</code> , <code>residentialunits</code> leer, <code>billingunitMscnumber</code>)	Datenqualität im ERP prüfen
MISSING	Referenz auf nicht existierende Entität (selten bei BUD, eher bei OSR)	Mapping prüfen
NOT_EXPECTED	Feld oder Eintrag, der für diesen MSC nicht zulässig ist (z.B. <code>additionalname</code> bei MSC ohne Unterstützung)	Optionale Felder ggf. entfernen
IMPLAUSIBLE	Werte-Konflikte: <code>heatedfloorspace</code> > <code>totalfloorspace</code> , <code>validfrom</code> in ferner Zukunft, ungültiger Ländercode, ungültige Etagennummer	Daten korrigieren

`message` ist MSC-spezifisch und nicht standardisiert. `shortmessage` ist der verlässliche Routing-Key für programmgesteuerte Fehlerbehandlung im ERP.

9.3 Retry-Strategie

Netzwerkfehler / Timeout (keine HTTP-Antwort):

- Sofortiger Retry mit identischem Body (POST ist idempotent)
- Nach 3 Fehlversuchen: in Retry-Queue, erneuter Versuch nach 5 Min

HTTP 500:

- Exponentieller Backoff: 1 Min, 5 Min, 15 Min, 1 Std
- Nach 4 Versuchen: manuelle Eskalation + MSC kontaktieren

HTTP 400 (Validierungsfehler):

- KEIN automatischer Retry
- Fehlermeldung im ERP anzeigen, manueller Korrekturworkflow

HTTP 401:

- Token erneuern, einmaliger Retry
- Kein weiterer Retry bei erneutem 401

HTTP 409 (Setup wird bereits bearbeitet):

- KEIN sofortiger Retry mit identischem Body
- AE bleibt IN_PROGRESS: über regelmäßigen Status-Check auf COMPLETED warten
- Erst nach COMPLETED (bzw. erneutem OPEN) ggf. neues Setup senden

HTTP 403 / 404:

- KEIN automatischer Retry
- Alarm auslösen, manuelle Klärung erforderlich

9.4 Häufige Fehler und ihre Ursachen

Symptom	Wahrscheinliche Ursache	Lösung
404 bei POST / setup/{billingunit}	billingunitMscnumber falsch oder AE noch nicht beim MSC freigeschaltet	GET /billingunits aufrufen, vorhandene Nummern verwenden
404 bei GET / setup/{billingunit}	Setup noch nicht abgeschlossen (Status OPEN / IN_PROGRESS) – es liegt noch kein Ergebnis vor	Status-Check gemäß §7.4; GET / setup erst bei COMPLETED aufrufen
409 bei POST / setup/{billingunit}	Für die AE läuft bereits ein Setup beim MSC (IN_PROGRESS); ein zweites POST wurde abgesetzt, bevor der erste Vorgang abgeschlossen war	Nicht erneut senden; Status-Check abwarten, bis die AE auf COMPLETED (oder erneut OPEN) wechselt
400 INCOMPLETE bei jedem POST	customerMscnumber oder residentialunits fehlen	Pflichtfelder gemäß §5.2 ergänzen
Setup bleibt dauerhaft IN_PROGRESS	MSC bearbeitet das Setup manuell (z.B. bei Datendifferenzen) – mehrere Tage sind dabei normal	Zunächst weiter über den täglichen Status-Check beobachten; erst bei auffällig langer Dauer MSC kontaktieren, transactionid nennen
additional[] enthält viele NE	PM-Daten unvollständig im Vergleich zum MSC-Bestand	Stammdatenabgleich mit MSC anstoßen
Wohnungen werden nach erneutem Setup nicht wiedererkannt	residentialunitPmnumber zwischen den Setups geändert	PM-Nummer stabil halten – niemals ändern
Länderkürzel DE abgelehnt	Muss ISO 3166-1 alpha-3 sein: DEU statt DE	Ländercode-Mapping im ERP anpassen
400 IMPLAUSIBLE bei Flächen	heatedfloorspace > totalfloorspace	Flächendaten im ERP korrigieren
Etagennummer abgelehnt	Wert außerhalb 1–999 oder negativ	Etagenmapping im ERP prüfen

10. Logging-Empfehlungen

Folgende Daten je BUD-Aufruf in einer audit-sicheren Log-Datenstruktur speichern:

Feld	Pflicht	Zweck
Zeitstempel (ISO 8601)	Ja	Reihenfolge der Setup-Versionen rekonstruieren
HTTP-Methode + Pfad	Ja	GET vs. POST, Listing vs. Setup unterscheiden
<code>billingunitMscnumber</code>	Ja (bei <code>/setup</code>)	Nachvollziehbarkeit, MSC-Support
<code>customerMscnumber</code>	Empfohlen	Mandantenzuordnung
HTTP-Status	Ja	Fehleranalyse
<code>transactionid</code> aus Response	Ja (bei 200/400 mit ID)	MSC-Support-Anfragen identifizieren
Anzahl <code>residentialunits</code> gesendet	Empfohlen	Vollständigkeitsprüfung
Setup-Status vor und nach Aufruf	Empfohlen	Workflow-Audit
Fehlermeldungen bei 400	Ja	Fehlerverfolgung
Anzahl <code>matched</code> / <code>additional</code> (bei GET /setup)	Empfohlen	Matching-Qualität nachvollziehen
Benutzer / Trigger-Quelle	Empfohlen	Wer/was hat den Request ausgelöst

11. Implementierungs-Checkliste

Setup

- ☐ OAuth 2.0 Client Credentials vom MSC erhalten
- ☐ `tokenUrl` des MSC konfiguriert
- ☐ Basis-URL des MSC konfiguriert
- ☐ `customerMscnumber` aus vertraglichem Onboarding bekannt
- ☐ HTTPS-Trust-Store vorbereitet (Zertifikate des MSC)

Datenmapping

- ☐ ERP-Liegenschaft → AE: `billingunitPmnumber` aus stabilem ERP-Schlüssel abgebildet
- ☐

ERP-Wohneinheit → `SetupRequestResidentialUnit` mit `residentialunitPmnumber` (stabil, mieterunabhängig)

- ☐ Adressen vollständig: Straße, PLZ, Stadt, ISO-3-Ländercode
- ☐ Flächen (Wohnfläche, beheizt/gesamt) numerisch mit Punkt-Dezimaltrennzeichen
- ☐ Etagentypen auf Enum (`UPPER_FLOOR` , `GROUND_FLOOR` , ...) gemappt
- ☐ Aktuelle Hauptmieter als `currentbillingrecipient` zum Matching mitgegeben
- ☐ Ländercodes auf ISO 3166-1 alpha-3 umgestellt (z.B. `DEU` statt `DE`)

Workflow

- ☐ `GET /billingunits` als Initial-Inventory implementiert
- ☐ Pagination (`offset` / `limit`) bei großen Mandaten implementiert
- ☐ Inkrementeller Abgleich über `from` -Parameter implementiert
- ☐ Status-Polling auf `setupstatus = COMPLETED` implementiert
- ☐ `GET /billingunits/setup/{...}` löst Übernahme der `residentialunitMscnumber` ins ERP aus
- ☐ `additional[]` aus Setup-Ergebnis wird dem Sachbearbeiter zur Klärung angezeigt
- ☐ Erneute Setup-Sendung bei NE-Strukturänderung im ERP implementiert

Konsistenzprüfungen (vor dem Senden)

- ☐ `billingunitMscnumber` im Body identisch mit Path-Parameter
- ☐ `customerMscnumber` befüllt
- ☐ Mindestens eine NE in `residentialunits[]`
- ☐ `heatedfloorspace` ≤ `totalfloorspace` in jeder NE (falls beide gepflegt)
- ☐ `heatedenclosedspace` ≤ `totalenclosedspace` in jeder NE (falls beide gepflegt)
- ☐ Keine doppelte `residentialunitPmnumber` innerhalb der AE

Fehlerbehandlung

- ☐ 200 → `transactionid` wird protokolliert, Setup-Status weiterverfolgt
- ☐ 400 → ValidationResponse ausgelesen, shortmessage ausgewertet, Sachbearbeiter-Anzeige implementiert
- ☐ 401 → Token-Refresh + einmaliger Retry implementiert
- ☐ 403 / 404 → Alarm ohne automatischen Retry
- ☐ 500 → Exponentieller Backoff implementiert
- ☐ Netzwerkfehler → idempotenter Retry implementiert

Tests

- ☐ GET `/billingunits` ohne Filter → 200 mit Liste
- ☐ GET `/billingunits?status=OPEN` → 200 nur mit offenen AE
- ☐ GET `/billingunits?from=YYYY-MM-DD` → 200 mit gefilterten AE
- ☐ Pagination: zwei aufeinanderfolgende Seiten → vollständige Liste rekonstruierbar
- ☐ POST `/setup/{...}` mit Minimal-Pflichtdaten → 200 + transactionid
- ☐ POST `/setup/{...}` mit vollständiger NE-Struktur → 200 + transactionid
- ☐ POST `/setup/{...}` mit fehlendem `customerMscnumber` → 400 + INCOMPLETE
- ☐ POST `/setup/{...}` mit Path-/Body-Inkonsistenz → 400
- ☐ POST `/setup/{...}` mit unbekannter AE → 404
- ☐ GET `/setup/{...}` nach Abschluss → 200 mit `matched[]` und ggf. `additional[]`
- ☐ GET `/setup/{...}` vor Abschluss (Status `OPEN` / `IN_PROGRESS`) → 404 (kein Ergebnis vorhanden)
- ☐ Token-Ablauf → Token-Refresh und Retry erfolgreich

12. Testempfehlungen

Testumgebung

Für Integrationstests steht die **bved Testplattform** zur Verfügung. Sie bildet das Verhalten eines echten MSC nach und antwortet auf BUD-Anfragen mit realistischen Mock-Daten.

- **Testplattform / Swagger UI:** <https://bved-api.de/testplatform/swagger-ui/index.html>

Empfohlene Testszenarien

Test	Erwartetes Ergebnis
GET /billingunits ohne Filter	200 mit Liste aller AE des Test-Mandanten
GET /billingunits?status=OPEN	200 mit nur offenen AE
GET /billingunits?status=IN_PROGRESS	200 mit nur laufenden Setups
GET /billingunits?status=COMPLETED	200 mit nur abgeschlossenen Setups
GET /billingunits?from=YYYY-MM-DD	200 mit nur seit dem Stichtag aktualisierten AE
GET /billingunits?offset=0&limit=10 + ? offset=10&limit=10	Lücken-/duplikatfreie Gesamtliste
POST /billingunits/setup/{...} mit gültigem Body	200 + transactionid
POST /billingunits/setup/{...} ohne customerMscnumber	400 + INCOMPLETE
POST /billingunits/setup/{...} ohne residentialunits	400 + INCOMPLETE
POST /billingunits/setup/{...} mit heatedfloorspace > totalfloorspace	400 + IMPLAUSIBLE
POST /billingunits/setup/{...} mit Pfad-/Body-Inkonsistenz	400
POST /billingunits/setup/{...} mit unbekannter AE	404
GET /billingunits/setup/{...} für abgeschlossenes Setup	200 mit matched[]
GET /billingunits/setup/{...} für laufendes Setup (nicht COMPLETED)	404 — kein Ergebnis vorhanden, solange das Setup nicht abgeschlossen ist

Test	Erwartetes Ergebnis
Token-Ablauf während Request	401 , nach Token-Refresh 200