API Schnittstellenspezifikation

- API-Anforderungen für Alarmierungen
- API-Anforderungen für Telemetrie-Upload
- API-Beschreibung für Telemetrie-Abruf (GET)

API-Anforderungen für Alarmierungen

Version: 1.2
Stand: März 2026
Herausgeber: Dexa Solutions GmbH
Produkt: Safe Fire House (SFH)

---

1. Übersicht

Diese Spezifikation definiert die REST-API-Schnittstelle für die Übermittlung von Alarmdaten von der Safe Fire House Brandwarnanlage an externe Alarmierungsdienste. Die API ermöglicht sowohl die Erstmeldung eines Alarms als auch nachfolgende Updates bei Broadcast-Alarmen.

1.1 Alarmtypen

Typ	Beschreibung	Methode
ALARM	Lokaler Rauchalarm – Einzelner Rauchsensor hat ausgelöst	POST
ALARM (Broadcast)	Folge-Alarm – Weitere Rauchsensoren in Funkreichweite	PUT
TEST	Schnittstellen-Test zur Validierung der Verbindung	POST

---

2. API-Parameter

Parameter	Wert
Base-URL	**https://{partner-domain}/api/v1**
Content-Type	**application/json; charset=UTF-8**
Accept	**application/json**
Zeichenkodierung	UTF-8

2.1 Authentifizierung

Die Authentifizierung erfolgt via **Bearer Token** im HTTP-Header:

```
Authorization: Bearer {access_token}
```

Parameter	Beschreibung
access_token	Vom Partner bereitgestellter API-Schlüssel (min. 32 Zeichen)

Hinweis: Der Token wird pro Kunde/Standort vom Partner generiert und im SFH-System hinterlegt.

2.2 Rate Limiting

Parameter	Wert
Max. Requests	60 pro Minute
Retry-After	Bei HTTP 429 im Header angegeben

---

3. Erstalarm (POST)

Sendet einen neuen Alarm an das Partner-System. Der Partner legt einen neuen Alarm-Datensatz an und gibt eine eindeutige **`alarmId`** zurück.

3.1 Request

```
POST /api/v1/alarms HTTP/1.1
Host: {partner-domain}
Authorization: Bearer {access_token}
Content-Type: application/json
Accept: application/json
```

3.2 Request-Body

```
{
 "externalCreatedAt": "2026-03-13T15:30:00Z",
 "externalId": "SFH-20260313-153000-001",
 "keyword": "ALARM",
 "keywordAddition": "RAUCHSENSOR-ALARM",
 "info": "Rauchsensor hat durch lokale Raucherkennung ausgelöst!",
 "priority": false,
 "send_push": true,
 "send_sms": false,
 "send_call": false,
 "group": "FW-Musterstadt-Zug1",
 "destination": {
  "objectName": "Feuerwehrgerätehaus Musterstadt",
  "info": "Fahrzeughalle",
  "street": "Hauptstraße",
  "houseNumber": "112",
  "zipCode": "12345",
  "city": "Musterstadt",
```

```
    "coordinates": {
      "latitude": 51.123456,
      "longitude": 7.654321
    },
    "fireAlarmSystem": "Safe Fire House"
  },
  "publisherInfos": {
    "systemName": "DXO-SFH-CU-X-02",
    "version": "2.0"
  },
  "reporter": [
    {
      "name": "HLF20-Kabine",
      "info": "Raucherkennung"
    }
  ]
}
```


3.3 Request-Felder

Root-Objekt

Feld	Typ	Pflicht	Beschreibung
externalCreatedAt	string	?	Zeitstempel der Alarmerstellung (ISO 8601, UTC)
externalId	string	?	Eindeutige Alarm-ID aus dem SFH-System (für Idempotenz)
keyword	string	?	Alarmtyp: **ALARM** oder **TEST**
keywordAddition	string	?	Detailbeschreibung: **RAUCHSENSOR-ALARM**, **SCHNITTSTELLEN-TEST**
info	string	?	Freitext-Information zum Alarm
priority	boolean	?	Prioritäts-Flag (reserviert für zukünftige Nutzung)

Feld	Typ	Pflicht	Beschreibung
send_push	boolean	?	Push-Benachrichtigung senden
send_sms	boolean	?	SMS-Benachrichtigung senden
send_call	boolean	?	Telefonanruf auslösen
group	string	?	Alarmierungsgruppe/RIC beim Partner
destination	object	?	Standort-Objekt (siehe unten)
publisherInfos	object	?	System-Informationen (siehe unten)
reporter	array	?	Array von Rauchsensor-Objekten (siehe unten)

destination-Objekt

Feld	Typ	Pflicht	Beschreibung
objectName	string	?	Name des Gebäudes/Objekts
info	string		Zusatzinformation zum Standort
street	string	?	Straßenname
houseNumber	string	?	Hausnummer
zipCode	string	?	Postleitzahl
city	string	?	Stadt/Ort
coordinates	object	?	Koordinaten-Objekt mit **latitude** und **longitude**
fireAlarmSystem	string	?	Systemkennung, immer **Safe Fire House**

coordinates-Objekt

Feld	Typ	Pflicht	Beschreibung
latitude	number	?	Breitengrad (WGS84, Dezimalgrad)
longitude	number	?	Längengrad (WGS84, Dezimalgrad)

publisherInfos-Objekt

Feld	Typ	Pflicht	Beschreibung
systemName	string	?	Produktkennung, z.B. **DXO-SFH-CU-X-02**
version	string	?	Produktversion: **1.0** oder **2.0**

reporter-Objekt (Array-Element)

Feld	Typ	Pflicht	Beschreibung
name	string	?	Gerätename/OPTA des auslösenden Rauchsensors
info	string	?	Art der Erkennung, z.B. **Raucherkennung**

3.4 Response (Erfolg)

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "status": "created",
  "alarmId": "550e8400-e29b-41d4-a716-446655440000",
  "received": "2026-03-13T15:30:01Z"
}
```

Feld	Typ	Beschreibung
status	string	**created** bei erfolgreichem Anlegen
alarmId	string	**Eindeutige ID des angelegten Alarms** (UUID oder PK) – wird für PUT benötigt!
received	string	Zeitstempel der Verarbeitung beim Partner (ISO 8601, UTC)

---

4. Alarm-Update (PUT)

Aktualisiert einen bestehenden Alarm (z.B. bei Broadcast-Alarm, wenn weitere Rauchsensoren auslösen). Die **alarmId** aus der POST-Response wird im URL-Pfad übergeben.

4.1 Request

```
PUT /api/v1/alarms/{alarmId} HTTP/1.1
Host: {partner-domain}
Authorization: Bearer {access_token}
Content-Type: application/json
Accept: application/json
```

URL-Parameter:

Parameter	Beschreibung
`alarmId`	Die vom Partner beim POST zurückgegebene Alarm-ID

4.2 Request-Body

```
{
 "alarmId": "550e8400-e29b-41d4-a716-446655440000",
 "externalId": "SFH-20260313-153000-001",
 "externalUpdatedAt": "2026-03-13T15:30:10Z",
 "keyword": "ALARM",
 "keywordAddition": "RAUCHSENSOR-ALARM (BROADCAST)",
 "info": "Weitere Rauchsensoren haben durch Broadcast-Alarm ausgelöst!",
 "priority": false,
 "send_push": true,
 "send_sms": false,
 "send_call": false,
 "group": "FW-Musterstadt-Zug1",
 "destination": {
  "objectName": "Feuerwehrgerätehaus Musterstadt",
  "info": "Fahrzeughalle",
  "street": "Hauptstraße",
  "houseNumber": "112",
  "zipCode": "12345",
  "city": "Musterstadt",
  "coordinates": {
   "latitude": 51.123456,
   "longitude": 7.654321
  },
```

```
  "fireAlarmSystem": "Safe Fire House"
 },
 "publisherInfos": {
  "systemName": "DXO-SFH-CU-X-02",
  "version": "2.0"
 },
 "reporter": [
  {
   "name": "HLF20-Kabine",
   "info": "Raucherkennung"
  },
  {
   "name": "HLF20-Mannschaftsraum",
   "info": "Raucherkennung (Broadcast)"
  }
 ]
}
```

4.3 Unterschiede zum POST

Feld	POST	PUT
alarmId	Nicht vorhanden	? Pflicht (im Body UND URL)
externalCreatedAt	?	Nicht vorhanden
externalUpdatedAt	Nicht vorhanden	? Pflicht
keywordAddition	**RAUCHSENSOR-ALARM**	**RAUCHSENSOR-ALARM (BROADCAST)**
reporter	1 Rauchsensor	1+ Rauchsensoren (kumulativ)

4.4 Response (Erfolg)

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
 "status": "updated",
```

```
  "alarmId": "550e8400-e29b-41d4-a716-446655440000",
  "received": "2026-03-13T15:30:11Z"
}
```

---

5. Alarm-Status abrufen (GET)

Ruft den aktuellen Status eines Alarms ab, einschließlich der Rückmeldungen der alarmierten Einsatzkräfte.

5.1 Request

```
GET /api/v1/alarms/{alarmId} HTTP/1.1
Host: {partner-domain}
Authorization: Bearer {access_token}
Accept: application/json
```

URL-Parameter:

Parameter	Beschreibung
`alarmId`	Die vom Partner beim POST zurückgegebene Alarm-ID

5.2 Response (Erfolg)

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "status": "ok",
  "alarmId": "550e8400-e29b-41d4-a716-446655440000",
  "externalId": "SFH-20260313-153000-001",
  "alarmStatus": "active",
  "createdAt": "2026-03-13T15:30:01Z",
  "updatedAt": "2026-03-13T15:30:11Z",
  "feedback": {
    "total": 12,
    "responses": [
      {
        "type": "coming",
        "label": "Komme",
```

```
      "count": 7
    },
    {
      "type": "coming_delayed",
      "label": "Komme später",
      "count": 2,
      "details": [
        { "eta": 5, "count": 1 },
        { "eta": 10, "count": 1 }
      ]
    },
    {
      "type": "not_available",
      "label": "Nicht verfügbar",
      "count": 3
    }
  ],
  "pending": 5
 },
 "received": "2026-03-13T15:31:00Z"
}
```


5.3 Response-Felder

Root-Objekt

Feld	Typ	Beschreibung
status	string	**ok** bei erfolgreicher Abfrage
alarmId	string	Partner-interne Alarm-ID (UUID/PK)
externalId	string	Ursprüngliche ID aus dem SFH-System (für Abgleich)
alarmStatus	string	Aktueller Alarmstatus (siehe Enum)
createdAt	string	Zeitpunkt der Alarmerstellung beim Partner (ISO 8601)
updatedAt	string	Zeitpunkt der letzten Aktualisierung (ISO 8601)
feedback	object	Rückmeldungs-Objekt (siehe unten)

Feld	Typ	Beschreibung
received	string	Zeitstempel dieser Abfrage (ISO 8601)

alarmStatus Enum

Wert	Beschreibung
active	Alarm ist aktiv, Alarmierung läuft
acknowledged	Alarm wurde quittiert
closed	Alarm wurde abgeschlossen
cancelled	Alarm wurde storniert

feedback-Objekt

Feld	Typ	Beschreibung
total	integer	Gesamtzahl der alarmierten Einsatzkräfte
responses	array	Array von Rückmelde-Objekten (siehe unten)
pending	integer	Anzahl noch ausstehender Rückmeldungen

responses-Objekt (Array-Element)

Feld	Typ	Pflicht	Beschreibung
type	string	?	Maschinenlesbarer Rückmeldetyp (siehe Enum)
label	string	?	Menschenlesbarer Text (Sprache des Partners)
count	integer	?	Anzahl der Rückmeldungen dieses Typs
details	array		Optional: Detailierte Aufschlüsselung (z.B. ETA-Zeiten)

response.type Enum (Standardisiert)

Typ	Beschreibung
coming	Kommt zum Einsatz
coming_delayed	Kommt später (mit ETA)

Typ	Beschreibung
not_available	Nicht verfügbar / Kommt nicht
standby	Bereitschaft / Evtl. verfügbar
on_scene	Bereits vor Ort
unknown	Sonstiger/Unbekannter Status

details-Objekt (Optional, für coming_delayed)

Feld	Typ	Beschreibung
eta	integer	Geschätzte Ankunftszeit in Minuten
count	integer	Anzahl mit dieser ETA

5.4 Beispiel: Status-Abfrage

```
curl -X GET "https://partner.example.com/api/v1/alarms/550e8400-e29b-41d4-a716-446655440000" \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIs..." \
  -H "Accept: application/json"
```

Response:

```
{
  "status": "ok",
  "alarmId": "550e8400-e29b-41d4-a716-446655440000",
  "externalId": "SFH-20260313-153000-001",
  "alarmStatus": "active",
  "createdAt": "2026-03-13T15:30:01Z",
  "updatedAt": "2026-03-13T15:30:11Z",
  "feedback": {
    "total": 12,
    "responses": [
      { "type": "coming", "label": "Komme", "count": 7 },
      { "type": "coming_delayed", "label": "Komme später", "count": 2,
"details": [{ "eta": 5, "count": 1 }, { "eta": 10, "count": 1 }] },
      { "type": "not_available", "label": "Nicht verfügbar", "count": 3 }
    ],
```

```
    "pending": 5
  },
  "received": "2026-03-13T15:31:00Z"
}
```

---

6. Fehlerbehandlung

6.1 HTTP-Statuscodes

Code	Bedeutung	Beschreibung
200	OK	Alarm erfolgreich aktualisiert (PUT)
201	Created	Alarm erfolgreich angelegt (POST)
400	Bad Request	Ungültiger Request-Body oder fehlende Pflichtfelder
401	Unauthorized	Fehlender oder ungültiger Bearer Token
403	Forbidden	Token gültig, aber keine Berechtigung für diese Ressource
404	Not Found	Alarm-ID nicht gefunden (bei PUT)
409	Conflict	Alarm mit dieser **externalId** existiert bereits (bei POST)
429	Too Many Requests	Rate Limit überschritten
500	Internal Server Error	Serverfehler beim Partner
503	Service Unavailable	Partner-System temporär nicht verfügbar

6.2 Fehler-Response

```
{
  "status": "error",
  "error": "invalid_payload",
  "message": "Field 'externalId' is required",
  "received": "2026-03-13T15:30:01Z"
}
```

Feld	Typ	Beschreibung
status	string	Immer **error**

Feld	Typ	Beschreibung
error	string	Fehlercode (siehe unten)
message	string	Menschenlesbare Fehlerbeschreibung
received	string	Zeitstempel der Fehlerverarbeitung

6.3 Fehlercodes

Code	Beschreibung
invalid_payload	JSON-Syntax ungültig oder Pflichtfeld fehlt
invalid_field	Feldwert entspricht nicht dem erwarteten Format
unauthorized	Token fehlt oder ist ungültig
forbidden	Keine Berechtigung für diese Operation
not_found	Ressource (Alarm) nicht gefunden
duplicate	Alarm mit dieser **externalId** existiert bereits
rate_limited	Zu viele Anfragen
internal_error	Interner Serverfehler

---

7. Idempotenz & Retry-Verhalten

7.1 Idempotenz

Das Feld **externalId** dient der Idempotenz-Sicherung:

- Bei wiederholtem POST mit gleicher **externalId** sollte der Partner **HTTP 409 Conflict** zurückgeben
- Alternativ kann der Partner ein Upsert-Verhalten implementieren (Update statt Insert)

7.2 Retry-Strategie (SFH-seitig)

Fehlertyp	Retry	Wartezeit
Netzwerkfehler	Ja	5s, 10s, 30s
HTTP 5xx	Ja	5s, 10s, 30s
HTTP 429	Ja	Retry-After Header beachten
HTTP 4xx (außer 429)	Nein	–

---

8. Sicherheitsanforderungen

Anforderung	Beschreibung
Transport	Ausschließlich HTTPS (TLS 1.2+)
Token-Speicherung	Access Token verschlüsselt auf der SFH-Zentrale
Token-Rotation	Empfohlen: Jährliche Erneuerung
IP-Whitelisting	Optional: Partner kann SFH-IPs whitelisten

---

9. Beispiel: Vollständiger Alarm-Flow

9.1 Schritt 1: Lokaler Alarm (POST)

Ein Rauchsensor im HLF20 löst aus:

```
curl -X POST "https://partner.example.com/api/v1/alarms" \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIs..." \
  -H "Content-Type: application/json" \
  -d '{
    "externalCreatedAt": "2026-03-13T15:30:00Z",
    "externalId": "SFH-20260313-153000-001",
    "keyword": "ALARM",
    "keywordAddition": "RAUCHSENSOR-ALARM",
    "info": "Rauchsensor hat durch lokale Raucherkennung ausgelöst!",
    "priority": false,
    "send_push": true,
    "send_sms": false,
    "send_call": false,
    "group": "FW-Musterstadt-Zug1",
    "destination": {
      "objectName": "Feuerwehrgerätehaus Musterstadt",
      "info": "Fahrzeughalle",
      "street": "Hauptstraße",
      "houseNumber": "112",
      "zipCode": "12345",
      "city": "Musterstadt",
      "coordinates": { "latitude": 51.123456, "longitude": 7.654321 },
      "fireAlarmSystem": "Safe Fire House"
    },
```

```
  "publisherInfos": { "systemName": "DXO-SFH-CU-X-02", "version": "2.0" },
  "reporter": [{ "name": "HLF20-Kabine", "info": "Raucherkennung" }]
 }'
```

Response:

```
{
 "status": "created",
 "alarmId": "550e8400-e29b-41d4-a716-446655440000",
 "received": "2026-03-13T15:30:01Z"
}
```

9.2 Schritt 2: Broadcast-Alarm (PUT)

~10 Sekunden später lösen weitere Rauchsensoren im Fahrzeug aus:

```
curl -X PUT "https://partner.example.com/api/v1/alarms/550e8400-e29b-41d4-a716-446655440000" \
 -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIs..." \
 -H "Content-Type: application/json" \
 -d '{
  "alarmId": "550e8400-e29b-41d4-a716-446655440000",
  "externalId": "SFH-20260313-153000-001",
  "externalUpdatedAt": "2026-03-13T15:30:10Z",
  "keyword": "ALARM",
  "keywordAddition": "RAUCHSENSOR-ALARM (BROADCAST)",
  "info": "Weitere Rauchsensoren haben durch Broadcast-Alarm ausgelöst!",
  "priority": false,
  "send_push": true,
  "send_sms": false,
  "send_call": false,
  "group": "FW-Musterstadt-Zug1",
  "destination": {
   "objectName": "Feuerwehrgerätehaus Musterstadt",
   "info": "Fahrzeughalle",
   "street": "Hauptstraße",
   "houseNumber": "112",
   "zipCode": "12345",
```

```
    "city": "Musterstadt",
    "coordinates": { "latitude": 51.123456, "longitude": 7.654321 },
    "fireAlarmSystem": "Safe Fire House"
   },
   "publisherInfos": { "systemName": "DXO-SFH-CU-X-02", "version": "2.0" },
   "reporter": [
    { "name": "HLF20-Kabine", "info": "Raucherkennung" },
    { "name": "HLF20-Mannschaftsraum", "info": "Raucherkennung
(Broadcast)" }
   ]
 }'
```

Response:

```
{
 "status": "updated",
 "alarmId": "550e8400-e29b-41d4-a716-446655440000",
 "received": "2026-03-13T15:30:11Z"
}
```

Anhang A: Constraints

Feld	Constraint
externalId	Max. 50 Zeichen, Pattern: **[A-Za-z0-9\-]+**
keyword	Enum: **ALARM**, **TEST**
keywordAddition	Max. 50 Zeichen
info	Max. 500 Zeichen
group	Max. 100 Zeichen
destination.street	Max. 100 Zeichen
destination.houseNumber	Max. 10 Zeichen
destination.zipCode	5 Zeichen (DE)
destination.city	Max. 100 Zeichen

Feld	Constraint
coordinates.latitude	-90.0 bis 90.0
coordinates.longitude	-180.0 bis 180.0
reporter[].name	Max. 50 Zeichen
reporter Array	Min. 1 Element

API-Anforderungen für Telemetrie-Upload

Version: 1.1
Stand: März 2026
Herausgeber: Dexa Solutions GmbH
Produkt: Safe Fire House (SFH)

---

1. Übersicht

Dieses Dokument beschreibt die REST-API-Anforderungen für die Übermittlung von Telemetriedaten der Safe Fire House Brandwarnanlage. Die Zentrale sendet stündlich die aktuellen Daten der Anlage an einen REST-API-Endpoint.

Folgende Parameter müssen unterstützt werden:

Parameter	Wert
Method	**`POST`**
Content-Type	**`application/json`**
Accept	**`application/json`**
Frequenz	Zyklisch (stündlich) 24/7, azyklisch bei Alarm
Rate Limit	Max. 60 Requests/Minute

1.1 Authentifizierung

Eine der folgenden Authentifizierungsmethoden muss unterstützt werden:

Methode	Header / Mechanismus	Beispiel
API-Key	**`X-API-Key`**	**`X-API-Key: sk_live_abc123...`**
Bearer Token (JWT)	**`Authorization: Bearer`**	**`Authorization: Bearer eyJhbGciOiJIUzI1NiIs...`**
X.509 Client-Zertifikat	mTLS (Mutual TLS)	Client-Zertifikat im TLS-Handshake

---

2. Payload-Struktur

Root
├── timestamp
├── fireStation
├── deviceId
└── vehicles[]

```
├── vehicleId
├── sign
├── callSign
├── vehicleType
└── smokeDetectors[]
    ├── name
    ├── address
    ├── type
    └── ...
```

3. Root-Objekt

Key	Description	Type	Constraints
timestamp	Zeitstempel der Erstellung	**string**	ISO 8601 UTC (**YYYY-MM-DDTHH:mm:ssZ**)
fireStation	Wache (Name, Adresse)	**string**	Max. 150 Zeichen
deviceId	Seriennummer der Zentrale	**string**	14 Zeichen, hexadezimal
vehicles	Auflistung der Fahrzeuge	**array**	Array von Vehicle-Objekten

Beispiel:

```
{
  "timestamp": "2026-03-13T11:24:13Z",
  "fireStation": "Feuerwehr Feuerstadt, Hauptstr. 112, 01234 Feuerstadt",
  "deviceId": "001A2B3C4D5E6F",
  "vehicles": [ ... ]
}
```

4. Vehicle-Objekt

Key	Description	Type	Constraints
vehicleId	Fahrzeug-Identifikationsnummer (VIN)	**string**	17 Zeichen
sign	Kennzeichen	**string**	Max. 10 Zeichen

Key	Description	Type	Constraints
callSign	Funkrufname	**string**	Max. 50 Zeichen
vehicleType	Fahrzeugtyp	**string**	Max. 50 Zeichen
smokeDetectors	Auflistung der Rauchsensoren	**array**	Array von SmokeDetector-Objekten

Beispiel:

```
{
  "vehicleId": "WVWZZZ3CZWE123456",
  "sign": "FS-FW 112",
  "callSign": "1-HLF20-1",
  "vehicleType": "HLF20",
  "smokeDetectors": [ ... ]
}
```

---

5. SmokeDetector-Objekt

Key	Description	Type	Constraints
name	Rauchsensorbezeichnung	**string**	Max. 30 Zeichen
address	Rauchsensoradresse	**string**	14 Zeichen, hexadezimal
type	Rauchsensortyp	**string**	Max. 20 Zeichen
version	Hardware-Version	**integer**	? 1
firmware	Firmware-Version	**string**	Max. 9 Zeichen, Pattern: **[0-9.]+**
group	Gruppierung	**string**	**0** – **9** oder leer
teams	Reserviert	**array**	—
rssiDevice	Funkempfangswert Gerät (dBm)	**integer**	?128 bis 128
rssiPeer	Funkempfangswert Sender (dBm)	**integer**	?128 bis 128
unreachState	Flag: Gerät nicht erreichbar	**boolean**	**true** / **false**
unreachCumulative	Kumulierte Nichterreichbarkeit (Tage)	**integer**	0–9999

Key	Description	Type	Constraints
operationtime	Betriebszeit (Tage)	**integer**	0–9999
battery	Flag: Batterieleistung niedrig	**boolean**	**true** / **false**
voltage	Batteriespannung (V)	**float**	0.0–3.2
errorcode	Fehlercode	**integer**	0–99
alarmstate	Alarmstatus	**integer**	0–3 (siehe Enum)
smokelevel	Raucherkennungsgrad (%)	**float**	0.0–100.0
dirtlevel	Verschmutzungsgrad (%)	**float**	0.0–100.0
chamber	Flag: Rauchkammer verschmutzt	**boolean**	**true** / **false**

Beispiel:


```
{
  "name": "1-HLF20-1 RM1",
  "address": "00AABBCCDDEE11",
  "type": "DXO-SFH-SD-XX-02",
  "version": 1,
  "group": "",
  "teams": [],
  "firmware": "1.0.6",
  "rssiDevice": -65,
  "rssiPeer": 0,
  "battery": false,
  "unreachState": false,
  "unreachCumulative": 0,
  "operationtime": 180,
  "dirtlevel": 0.0,
  "smokelevel": 0.0,
  "alarmstate": 0,
  "voltage": 3.0,
  "chamber": false,
  "errorcode": 0
}
```

6. Enums

6.1 alarmstate

Wert	Bedeutung
0	Ruhezustand – Kein Rauch erkannt
1	Lokaler Alarm – Rauch erkannt
2	Reserviert
3	Broadcast Alarm – Anderer Sensor in Funkreichweite hat Rauch erkannt

---

7. Flag-Logik

Flag	Bedeutung wenn **true**	Zusatzinfo
chamber	Rauchkammer verschmutzt	Siehe **dirtlevel** (%)
battery	Batterieleistung niedrig	Siehe **voltage** (V)
unreachState	Gerät nicht erreichbar	Siehe **unreachCumulative** (Tage)

---

8. Response

8.1 Erwartete HTTP Status Codes

Code	Bedeutung
200 OK	Erfolgreich verarbeitet
400 Bad Request	Ungültiger Payload
401 Unauthorized	Fehlende oder ungültige Authentifizierung
403 Forbidden	Keine Berechtigung
429 Too Many Requests	Rate Limit überschritten
500 Internal Server Error	Serverfehler
503 Service Unavailable	Service nicht verfügbar

8.2 Success Response


```
{
 "status": "ok",
 "received": "2026-03-13T11:24:13Z"
}
```


8.3 Error Response


```
{
 "error": "invalid_payload",
 "message": "Field 'address' invalid"
}
```


---

9. Vollständiges Payload-Beispiel


```
{
 "timestamp": "2026-03-13T11:24:13Z",
 "fireStation": "Feuerwehr Feuerstadt, Hauptstr. 112, 01234 Feuerstadt",
 "deviceId": "001A2B3C4D5E6F",
 "vehicles": [
  {
   "vehicleId": "WVWZZZ3CZWE123456",
   "sign": "FS-FW 112",
   "callSign": "1-HLF20-1",
   "vehicleType": "HLF20",
   "smokeDetectors": [
    {
     "name": "1-HLF20-1 RM1",
     "address": "00AABBCCDDEE11",
     "type": "DXO-SFH-SD-XX-02",
     "version": 1,
     "group": "",
     "teams": [],
     "firmware": "1.0.6",
     "rssiDevice": -65,
     "rssiPeer": 0,
```

```
        "battery": false,
        "unreachState": false,
        "unreachCumulative": 0,
        "operationtime": 180,
        "dirtlevel": 0.0,
        "smokelevel": 0.0,
        "alarmstate": 0,
        "voltage": 3.0,
        "chamber": false,
        "errorcode": 0
      },
      {
        "name": "1-HLF20-1 RM2",
        "address": "00AABBCCDDEE22",
        "type": "DXO-SFH-SD-XX-02",
        "version": 1,
        "group": "",
        "teams": [],
        "firmware": "1.0.6",
        "rssiDevice": -72,
        "rssiPeer": 0,
        "battery": false,
        "unreachState": false,
        "unreachCumulative": 0,
        "operationtime": 180,
        "dirtlevel": 0.0,
        "smokelevel": 0.0,
        "alarmstate": 0,
        "voltage": 3.0,
        "chamber": false,
        "errorcode": 0
      }
    ]
  }
 ]
}
```

API-Beschreibung für Telemetrie-Abruf (GET)

Version: 1.5
Stand: Juni 2026
Herausgeber: Dexa Solutions GmbH
Produkt: Safe Fire House (SFH)

---

1. Übersicht

Dieses Dokument beschreibt den REST-API-Endpoint, über den ein externes System die aktuellen Telemetriedaten der Safe Fire House Brandwarnanlage **aktiv abruft** (Pull). Die Zentrale ist hier der **Server**, das abrufende System der **Client**. Die Antwort enthält den Datenbaum aus Wache, Fahrzeugen und Rauchsensoren.

Parameter	Wert
Method	**`GET`**
Pfad	**`/api/health/telemetry`** (mit oder ohne abschließenden **`/`**)
Accept	**`application/json`**
Response-Type	**`application/json; charset=utf-8`**
Frequenz	On-Demand (Client-gesteuert)
Rate Limit	5 Requests/Sekunde pro Client-IP, Burst 10 (sonst **`429`**)
Caching	Antwort bis zu 20 s serverseitig gecacht (siehe Abschnitt 9)

1.1 Authentifizierung

Der Endpoint erfordert ein **Bearer-Token** im **`Authorization`**-Header. Das Token wird je Zentrale vergeben und vertraulich an das abrufende System übergeben.

Methode	Header / Mechanismus	Beispiel
Bearer Token	**`Authorization: Bearer <token>`**	**`Authorization: Bearer bffdc50d5a1173159...`**

Fehlt der Header oder ist das Token ungültig, antwortet der Endpoint mit **`401`**. Ist serverseitig kein Token konfiguriert, antwortet er mit **`503`** (fail-closed, keine Datenausgabe).

---

2. Endpoint

`GET https://<zentrale-host>/api/health/telemetry`

- **<zentrale-host>** = IP oder Hostname der Zentrale im lokalen Netz (z. B. **192.168.1.16**).
- Erreichbar über den Reverse-Proxy der Zentrale; der **Authorization** -Header wird unverändert durchgereicht.
- Der abschließende Slash ist optional (**/api/health/telemetry** und **/api/health/telemetry/** sind gleichwertig).

---

3. Request

Es wird **kein** Request-Body gesendet. Erforderlicher Header:

Header	Pflicht	Wert
Authorization	ja	**Bearer <token>**

Beispiel:

```
curl --location 'https://192.168.1.16/api/health/telemetry/' \
  --header 'Authorization: Bearer <token>'
```

---

4. Payload-Struktur (Response)

```
Root
├── timestamp
├── fireStation
├── deviceId
└── objects[]
    ├── type            (vehicle | room | hall)
    ├── vehicleId
    ├── sign
    ├── callSign
    ├── vehicleType
    └── smokeDetectors[]
        ├── name
        ├── address
        ├── type
        └── ...
```

Hinweis zu Wertetypen: Alle Schlüssel sind camelCase. Alle skalaren Werte werden als JSON-String ausgegeben (auch Zahlen und Flags, z. B. `"rssiDevice": "-71"`, `"battery": "false"`, `"alarmState": "0"`). `objects` und `smokeDetectors` sind echte JSON-Arrays.

---

5. Root-Objekt

Key	Description	Type	Constraints
`timestamp`	Zeitstempel der Erstellung	**`string`**	ISO 8601 UTC (`YYYY-MM-DDTHH:mm:ssZ`)
`fireStation`	Wache (Name, Adresse)	**`string`**	Max. 150 Zeichen
`deviceId`	Seriennummer der Zentrale	**`string`**	14 Zeichen, hexadezimal
`objects`	Auflistung der Objekte (Fahrzeug/Raum/Halle)	**`array`**	Array von Objekt-Einträgen (siehe Abschnitt 6)

Beispiel:

```
{
 "timestamp": "2026-06-09T11:24:13Z",
 "fireStation": "Feuerwehr Feuerstadt, Hauptstr. 112, 01234 Feuerstadt",
 "deviceId": "001A2B3C4D5E6F",
 "objects": [ ... ]
}
```

---

6. Objekt-Eintrag

Ein Objekt-Eintrag bündelt die Rauchsensoren eines Trägers. Das Feld **`type`** unterscheidet die Träger-Art. Die fahrzeugspezifischen Felder (**`vehicleId`**, **`sign`**, **`callSign`**, **`vehicleType`**) sind bei **`type = "vehicle"`** befüllt; für **`room`** / **`hall`** können sie leer bzw. **`"n.a."`** sein.

Key	Description	Type	Constraints
`type`	Art des Trägers	**`string`**	Enum: **`"vehicle"`** | **`"room"`** | **`"hall"`** (derzeit nur **`"vehicle"`** belegt)

Key	Description	Type	Constraints
vehicleId	Fahrzeug-Identifikationsnummer (VIN)	**string**	17 Zeichen; **"n.a."** falls nicht hinterlegt (siehe 8.1)
sign	Kennzeichen	**string**	Max. 10 Zeichen; **"n.a."** falls nicht hinterlegt (siehe 8.1)
callSign	Funkrufname	**string**	Max. 50 Zeichen
vehicleType	Fahrzeugtyp	**string**	Max. 50 Zeichen; **"n.a."** falls nicht hinterlegt (siehe 8.1)
smokeDetectors	Auflistung der Rauchsensoren	**array**	Array von SmokeDetector-Objekten; **[]** falls keine Melder zugeordnet (siehe 8.2)

Beispiel:

```
{
 "type": "vehicle",
 "vehicleId": "WVWZZZ3CZWE123456",
 "sign": "FS-FW 112",
 "callSign": "1-HLF20-1",
 "vehicleType": "HLF20",
 "smokeDetectors": [ ... ]
}
```

---

7. SmokeDetector-Objekt

Alle Werte sind Strings (siehe Hinweis in Abschnitt 4). Fehlt ein einzelner Datapoint, wird ein typ-konformer Default geliefert (nie **null**) — siehe Abschnitt 8.3.

Key	Description	Type	Constraints
name	Rauchsensorbezeichnung	**string**	Max. 30 Zeichen
address	Rauchsensoradresse	**string**	14 Zeichen, hexadezimal
type	Rauchsensortyp	**string**	Konstant **"SFHSS02"**
version	Hardware-Version	**string**	numerisch, ? 1
group	Gruppierung	**string**	**0** – **9** oder leer
teams	Reserviert	**string**	i. d. R. leer

Key	Description	Type	Constraints
firmware	Firmware-Version	**string**	Max. 9 Zeichen, Pattern **[0-9.]+**
rssiDevice	Funkempfangswert Gerät (dBm)	**string**	numerisch, ?128 bis 128
rssiPeer	Funkempfangswert Sender (dBm)	**string**	numerisch, ?128 bis 128
battery	Flag: Batterieleistung niedrig	**string**	**"true"** / **"false"**
unreachState	Flag: Gerät nicht erreichbar	**string**	**"true"** / **"false"**
unreachCumulative	Kumulierte Nichterreichbarkeit (Tage)	**string**	numerisch (0–9999) oder **"n.a."** (siehe 8.2)
operationTime	Betriebszeit (Tage)	**string**	numerisch, 0–9999
dirtLevel	Verschmutzungsgrad	**string**	float-String (z. B. **"0.000000"**)
smokeLevel	Raucherkennungsgrad	**string**	float-String (z. B. **"0.000000"**)
alarmState	Alarmstatus	**string**	**"0"** – **"3"** (siehe Enum, Abschnitt 7.1)
voltage	Batteriespannung (V)	**string**	float-String (0.0–3.2)
chamber	Flag: Rauchkammer verschmutzt	**string**	**"true"** / **"false"**
errorCode	Fehlercode	**string**	numerisch, 0–99

Beispiel:


```
{
 "name": "1-HLF20-1 RM1",
 "address": "00AABBCCDDEE11",
 "type": "SFHSS02",
 "version": "1",
 "group": "",
 "teams": "",
 "firmware": "1.0.6",
 "rssiDevice": "-65",
 "rssiPeer": "0",
 "battery": "false",
```

```
  "unreachState": "false",
  "unreachCumulative": "0",
  "operationTime": "180",
  "dirtLevel": "0.000000",
  "smokeLevel": "0.000000",
  "alarmState": "0",
  "voltage": "3.000000",
  "chamber": "false",
  "errorCode": "0"
}
```

7.1 Enum: alarmState

Wert	Bedeutung
"0"	Ruhezustand – Kein Rauch erkannt
"1"	Lokaler Alarm – Rauch erkannt
"2"	Reserviert
"3"	Broadcast Alarm – Anderer Sensor in Funkreichweite hat Rauch erkannt

7.2 Flag-Logik

Flag	Bedeutung wenn "true"	Zusatzinfo
chamber	Rauchkammer verschmutzt	Siehe **dirtLevel**
battery	Batterieleistung niedrig	Siehe **voltage** (V)
unreachState	Gerät nicht erreichbar	Siehe **unreachCumulative** (Tage)

---

8. Sonderfälle & Defaults

8.1 Fahrzeug-Metadaten nicht deklariert

vehicleId, **sign** und **vehicleType** werden je **callSign** aus der Fahrzeug-Stammdatenpflege der Zentrale gelesen. Verhalten pro Feld (einzeln):

Situation	Ausgabe
callSign **fehlt** in der Stammdatenpflege	**"n.a."**
Eintrag vorhanden, **Wert leer**	**""** (leerer String)

Situation	Ausgabe
Eintrag + Wert vorhanden	der Wert

`callSign` selbst stammt aus der Fahrzeugliste und ist immer gesetzt.

8.2 Gerät nicht erreichbar / nicht gepairt

Fall	Verhalten
Melder **gepairt, aber offline**	erscheint im Baum; `unreachState` = `"true"`; `unreachCumulative` = Tage seit letztem Kontakt bzw. `"n.a."`; übrige Werte = **zuletzt bekannter Stand** (kein Live-Funk-Poll beim Abruf)
Melder **nicht (mehr) gepairt**	Melder fehlt im Array. Ein Fahrzeug ohne zugeordnete Melder liefert `"smokeDetectors": []`

> **`unreachCumulative` = `"n.a."` bedeutet „nicht in der Erreichbarkeits-Historie der Zentrale geführt", nicht zwingend „erreichbar".**

8.3 Fehlender Datapoint → typ-konformer Default

Existiert ein einzelner Sensor-Datapoint nicht (abweichendes Geräteprofil o. Ä.), wird statt `null` ein Default ausgegeben:

Feld(er)	Default
`battery`, `unreachState`, `chamber`	`"false"`
`rssiDevice`, `rssiPeer`, `errorCode`, `operationTime`, `alarmState`	`"0"`
`voltage`, `smokeLevel`, `dirtLevel`	`"0.000000"`
`firmware`, `group`, `version`, `teams` (keine Geräte-Metadaten)	`""`
`unreachCumulative` (keine Historie)	`"n.a."`

---

9. Caching & Nebenläufigkeit

- **TTL-Cache:** Die Antwort wird serverseitig bis zu **20 s** zwischengespeichert. Aufeinanderfolgende Abrufe innerhalb dieses Fensters liefern denselben (bis zu 20 s alten) Stand, ohne die Zentrale erneut abzufragen.

- **Single-Flight:** Pro Cache-Miss läuft höchstens **eine** Datenerhebung. Treffen mehrere Abrufe gleichzeitig ein, teilen sie sich das laufende Ergebnis; es werden keine parallelen Erhebungen gestartet.
- Die Werte spiegeln den zuletzt in der Zentrale bekannten Zustand der Sensoren wider (kein aktiver Funk-Poll der Geräte beim Abruf).

---

10. Response (HTTP)

10.1 Status Codes

Code	Bedeutung
200 OK	Telemetrie erfolgreich geliefert (Body = Datenbaum)
401 Unauthorized	Fehlender oder ungültiger **Authorization** -Header
429 Too Many Requests	Rate Limit überschritten
502 Bad Gateway	Telemetrie nicht lesbar (Zentrale nicht erreichbar)
503 Service Unavailable	Serverseitig kein API-Token konfiguriert

10.2 Success Response

Body ist der vollständige Telemetrie-Datenbaum (siehe Abschnitt 11).

10.3 Error Response

```
{
 "error": "unauthorized"
}
```

```
{
 "error": "telemetryUnavailable",
 "detail": "telemetry source returned HTTP 500"
}
```

error	HTTP	Bedeutung
unauthorized	**401**	Token fehlt/falsch
telemetryUnavailable	**502**	Datenerhebung fehlgeschlagen (**detail**)
apiTokenNotConfigured	**503**	Kein Token gesetzt

11. Vollständiges Response-Beispiel


```
{
 "timestamp": "2026-06-09T11:24:13Z",
 "fireStation": "Feuerwehr Feuerstadt, Hauptstr. 112, 01234 Feuerstadt",
 "deviceId": "001A2B3C4D5E6F",
 "objects": [
  {
   "type": "vehicle",
   "vehicleId": "WVWZZZ3CZWE123456",
   "sign": "FS-FW 112",
   "callSign": "1-HLF20-1",
   "vehicleType": "HLF20",
   "smokeDetectors": [
    {
     "name": "1-HLF20-1 RM1",
     "address": "00AABBCCDDEE11",
     "type": "SFHSS02",
     "version": "1",
     "group": "",
     "teams": "",
     "firmware": "1.0.6",
     "rssiDevice": "-65",
     "rssiPeer": "0",
     "battery": "false",
     "unreachState": "false",
     "unreachCumulative": "0",
     "operationTime": "180",
     "dirtLevel": "0.000000",
     "smokeLevel": "0.000000",
     "alarmState": "0",
     "voltage": "3.000000",
     "chamber": "false",
     "errorCode": "0"
    },
    {
     "name": "1-HLF20-1 RM2",
```

```
        "address": "00AABBCCDDEE22",
        "type": "SFHSS02",
        "version": "1",
        "group": "",
        "teams": "",
        "firmware": "1.0.6",
        "rssiDevice": "-72",
        "rssiPeer": "0",
        "battery": "false",
        "unreachState": "false",
        "unreachCumulative": "0",
        "operationTime": "180",
        "dirtLevel": "0.000000",
        "smokeLevel": "0.000000",
        "alarmState": "0",
        "voltage": "3.000000",
        "chamber": "false",
        "errorCode": "0"
      }
    ]
  }
 ]
}
```