

API Anforderungen für Alarmierungen

Version: 1.2 **Stand:** März 2026 **Herausgeber:** Dexa Consult 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

Parameter	Wert
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 Rauchererkennung 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": "Rauchererkennung"
  }
]
}

```

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)
send_push	boolean	?	Push-Benachrichtigung senden
send_sms	boolean	?	SMS-Benachrichtigung senden

Feld	Typ	Pflicht	Beschreibung
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": "Rauchererkennung"
  },
  {
    "name": "HLF20-Mannschaftsraum",
    "info": "Rauchererkennung (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
<code>alarmId</code>	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: Detaillierte Aufschlüsselung (z.B. ETA-Zeiten)

response.type Enum (Standardisiert)

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

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

details-Objekt (Optional, für `coming_delayed`)

Feld	Typ	Beschreibung
<code>eta</code>	integer	Geschätzte Ankunftszeit in Minuten
<code>count</code>	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 eyJhbGciOiJIUzI1NiIsInR5cCI6IHNpb20iLCJ1d2kiOiJ1dWkiLCJ1d2kiOiJ1dWkiLCJ1d2kiOiJ1dWki" \
  -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 },
      { "type": "not_available", "label": "Nicht verfügbar", "count": 3 }
    ]
  },
  "details": [
    { "eta": 5, "count": 1 },
    { "eta": 10, "count": 1 }
  ]
}
```

```

"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	–


```

"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": "Rauchererkennung" },
  { "name": "HLF20-Mannschaftsraum", "info": "Rauchererkennung
(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
coordinates.latitude	-90.0 bis 90.0

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