

API-Anforderungen für Gebäudezustandsmeldungen

Version: 1.0

Stand: 2026-10

Herausgeber: Dexa Solutions GmbH

Produkt: Safe Fire House (SFH) – Einbruchwarnsystem (EWS)

1. Übersicht

Diese Spezifikation beschreibt die REST-Schnittstelle, über die ein Drittsystem – zum Beispiel eine Zutrittskontrollanlage, eine Gebäudeleittechnik oder eine Hausautomation – dem Einbruchwarnsystem der Safe-Fire-House-Zentrale den **Gebäudezustand** meldet und den Zustand der Anlage abfragt.

Die Schnittstelle ersetzt den potentialfreien Kontakt eines Schlüsselschalters: Statt einen Draht zu schalten, teilt das Drittsystem der Zentrale über das Netzwerk mit, ob das Gebäude **zugeschlossen** oder **aufgeschlossen** ist.

1.1 Grundprinzip: Zustand statt Befehl

Das Drittsystem meldet ausschließlich einen **Zustand**, keinen Schaltbefehl. Ob daraus ein Scharf- oder Unscharfschalten folgt, entscheidet das Regelwerk der Zentrale – nach denselben Regeln wie beim Schlüsselschalter (Abschnitt 6).

Ein fehlerhaftes oder kompromittiertes Drittsystem kann dadurch höchstens einen falschen Gebäudezustand behaupten. Die Schutzregeln der Anlage – aktiver Alarm, Wartung, Störung – bleiben immer davor.

1.2 Funktionen

Funktion	Methode	Pfad	Recht
Zustand der Anlage abfragen	GET	<code>/api/intrusion/access/state</code>	Lesen
Gebäudezustand melden	POST	<code>/api/intrusion/access/report</code>	Schreiben

2. API-Parameter

Parameter	Wert
Adresse	<code>https://{adresse-der-zentrale}</code> – wird bei der Einrichtung mitgeteilt
Content-Type	<code>application/json; charset=UTF-8</code>
Accept	<code>application/json</code>
Zeichenkodierung	UTF-8
Max. Größe einer Anfrage	4 KB

Jede Zentrale hat **eine** Adresse, auch wenn sie mehrere Standorte überwacht. Welcher Standort gemeint ist, steht als Standort-Kennung in der Anfrage (Abschnitt 2.2).

Die Schnittstelle ist in der Voreinstellung nur im internen Netz des Kunden erreichbar. Ein Zugang von außen wird nur auf ausdrücklichen Wunsch eingerichtet.

2.1 Authentifizierung

Die Authentifizierung erfolgt über einen **Zugangsschlüssel (Token)** im HTTP-Kopf:

Authorization: Bearer {zugangsschlüssel}

Eigenschaft	Beschreibung
Ausgabe	Wird bei der Einrichtung auf der Zentrale erzeugt und einmalig angezeigt. Er kann danach nicht erneut angezeigt werden – geht er verloren, wird ein neuer erzeugt.
Speicherung	Die Zentrale speichert den Schlüssel nicht im Klartext, sondern nur einen daraus abgeleiteten Prüfwert.
Recht	Je Schlüssel festgelegt: Lesen oder Schreiben . Ein Schlüssel mit Recht „Schreiben“ darf auch abfragen.
Standorte	Je Schlüssel festgelegt, für welche Standorte er gilt.
Ablauf	Ohne Ablaufdatum (Voreinstellung) oder befristet, z. B. für Testzugänge.
Widerruf	Jederzeit auf der Zentrale möglich.
Absender	Optional auf bestimmte Absenderadressen beschränkbar.

Der Schlüssel gehört ausschließlich in den Kopf der Anfrage, niemals in die Adresse. Adressen werden in Protokollen gespeichert, der Kopf nicht.

Empfehlung: Für Systeme, die den Zustand nur anzeigen, einen Schlüssel mit Recht „Lesen“ verwenden. Wird ein solches System kompromittiert, kann die Anlage darüber nicht beeinflusst werden.

2.2 Standort-Kennung

Jeder Standort einer Zentrale hat eine Standort-Kennung (**locationHash** , 32 Zeichen hexadezimal). Sie wird bei der Einrichtung mitgeteilt.

Die Standort-Kennung ist ****keine**** Berechtigung. Sie sagt nur, welcher Standort gemeint ist. Ob er gemeldet oder abgefragt werden darf, entscheidet allein der Zugangsschlüssel. Eine Meldung für einen Standort, für den der Schlüssel nicht freigegeben ist, wird abgewiesen.

2.3 Rate Limiting

Parameter	Wert
Max. Anfragen	1 pro Sekunde je Absender, kurzzeitig bis zu 5
Bei Überschreitung	HTTP 429

Ein Zutrittssystem meldet bei Zustandswechsel und höchstens ein Lebenszeichen pro Minute. Mehr Anfragen sind im Betrieb nicht vorgesehen.

3. Zustand abfragen (GET)

Liefert den Zustand der Anlage für alle Standorte, für die der Schlüssel freigegeben ist.

3.1 Request

```
GET /api/intrusion/access/state HTTP/1.1  
Host: {adresse-der-zentrale}  
Authorization: Bearer {zugangsschlüssel}  
Accept: application/json
```

3.2 Response (Erfolg)

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

```
{  
  "locations": [  
    {  
      "locationHash": "9f2c4e7a1b3d5f6081a2c3d4e5f60718",  
      "name": "Lagerhalle Nord",  
      "armingState": "armed",  
      "alarmState": "none",  
      "fault": false,  
      "lastReportAt": "2026-10-14T06:58:12.000Z"  
    },  
    ],  
  "timestamp": "2026-10-14T07:02:40.000Z"  
}
```

3.3 Response-Felder

Root-Objekt

Feld	Typ	Beschreibung
<code>locations</code>	array	Ein Eintrag je freigegebenem Standort

Feld	Typ	Beschreibung
timestamp	string	Zeitpunkt der Antwort (ISO 8601, UTC)

locations-Objekt (Array-Element)

Feld	Typ	Beschreibung
locationHash	string	Standort-Kennung
name	string	Name des Standorts
armingState	string	Scharfzustand der Anlage (siehe Enum)
alarmState	string	Alarmzustand der Anlage (siehe Enum)
fault	boolean	true , wenn an diesem Standort eine Störung vorliegt – der Schnittstelle oder der Anlage (z. B. Kamera, Netzwerk)
lastReportAt	string null	Zeitpunkt der letzten angenommenen Meldung (ISO 8601, UTC), null wenn noch keine Meldung einging

armingState Enum

Wert	Beschreibung
armed	Standort ist scharf
arming	Scharfschaltung läuft (Vorlaufzeit, Kameras starten)
disarmed	Standort ist unscharf

alarmState Enum

Wert	Beschreibung
none	Kein Alarm
prealarm	Voralarm läuft
alarm	Alarm ist ausgelöst und noch nicht quittiert

Scharfzustand und Alarmzustand sind bewusst zwei getrennte Felder. Der Zustand „unscharf, Alarm noch offen“ entsteht, wenn bei laufendem Alarm aufgeschlossen wird (Abschnitt 6).

Den Gebäudezustand selbst (auf- oder zugeschlossen) liefert die Abfrage **nicht** zurück – diesen kennt das Drittsystem selbst.

4. Gebäudezustand melden (POST)

Meldet der Zentrale, ob das Gebäude eines Standorts zugeschlossen oder aufgeschossen ist. Erfordert einen Schlüssel mit Recht „Schreiben“.

4.1 Request

```
POST /api/intrusion/access/report HTTP/1.1
Host: {adresse-der-zentrale}
Authorization: Bearer {zugangsschlüssel}
Content-Type: application/json
Accept: application/json
```

4.2 Request-Body

```
{
  "locationHash": "9f2c4e7a1b3d5f6081a2c3d4e5f60718",
  "buildingLocked": true,
  "timestamp": "2026-10-14T06:58:12Z"
}
```

4.3 Request-Felder

Feld	Typ	Pflicht	Beschreibung
locationHash	string	?	Standort-Kennung des gemeldeten Standorts
buildingLocked	boolean	?	true = Gebäude zugeschlossen, false = Gebäude aufgeschossen
timestamp	string		Zeitpunkt der Meldung (ISO 8601). Empfohlen – aktiviert den Wiederholschutz (Abschnitt 7.3)

`buildingLocked` muss ein echter Wahrheitswert sein: `true` oder `false`. Zeichenketten wie `"true"`, Zahlen wie `1` oder fehlende Werte werden mit HTTP 400 abgewiesen.

4.4 Response (Erfolg)

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "ok": true,
  "accepted": { "buildingLocked": true },
  "timestamp": "2026-10-14T06:58:12.000Z"
}
```

Feld	Typ	Beschreibung
ok	boolean	true , wenn die Meldung angenommen wurde
accepted.buildingLocked	boolean	Der angenommene Gebäudezustand
timestamp	string	Zeitpunkt der Verarbeitung (ISO 8601, UTC)

„Angenommen“ heißt: Die Zentrale kennt jetzt den neuen Gebäudezustand. Ob die Anlage daraufhin geschaltet hat, zeigt die Abfrage (Abschnitt 3) – die Scharfschaltung kann je nach Regelwerk ausbleiben oder mit Vorlaufzeit erfolgen.

5. Fehlerbehandlung

5.1 HTTP-Statuscodes

Code	Bedeutung	Beschreibung
200	OK	Abfrage beantwortet bzw. Meldung angenommen
400	Bad Request	Ungültiger Inhalt, buildingLocked kein Wahrheitswert, oder timestamp unlesbar, zu alt bzw. bereits verwendet
401	Unauthorized	Schlüssel fehlt oder ist unbekannt
403	Forbidden	Schlüssel widerrufen oder abgelaufen, Absender nicht freigegeben, Standort für diesen Schlüssel nicht freigegeben, Schlüssel nur lesend, oder Anfrage nicht verschlüsselt (http statt https)
404	Not Found	Unbekannter Pfad

Code	Bedeutung	Beschreibung
413	Payload Too Large	Anfrage größer als 4 KB
429	Too Many Requests	Rate Limit überschritten
503	Service Unavailable	Schnittstelle auf der Zentrale vorübergehend nicht verfügbar

5.2 Fehler-Response

```
{
  "error": "forbidden"
}
```

Fehlercode	Beschreibung
unauthorized	Zugang abgewiesen (Schlüssel fehlt, unbekannt, widerrufen, abgelaufen oder Absender nicht freigegeben)
read_only	Schlüssel hat nur Leserecht
forbidden	Standort für diesen Schlüssel nicht freigegeben
bad_request	Inhalt ungültig
not_found	Unbekannter Pfad
unavailable	Schnittstelle vorübergehend nicht verfügbar

Die Fehlermeldungen nach außen sind bewusst knapp und nennen nicht, woran eine Anmeldung gescheitert ist. Der genaue Grund steht im Prüfprotokoll der Zentrale und kann vom Errichter bzw. von dexa eingesehen werden.

5.3 Retry-Strategie (Drittsystem)

Fehlertyp	Wiederholen	Wartezeit
Netzwerkfehler	Ja	5 s, 10 s, 30 s
HTTP 503	Ja	5 s, 10 s, 30 s
HTTP 429	Ja	mindestens 1 s
HTTP 400, 401, 403, 404	Nein	– Konfiguration prüfen

Bei einer Wiederholung mit Zeitstempel bitte einen **neuen** Zeitstempel verwenden – eine Anfrage mit bereits verwendetem Zeitstempel wird als Wiederholung abgewiesen.

6. Verhalten der Anlage

Eine angenommene Meldung wirkt wie das Schalten eines Schlüsselschalters:

Meldung	Wirkung
Aufgeschlossen (false)	Standort wird sofort unscharf geschaltet, ohne Vorlaufzeit
Zugeschlossen (true)	Standort wird über den normalen Weg scharf geschaltet: Vorlaufzeit, Kamerastart, Kameraprüfung
Gleicher Zustand wie zuvor	Keine Wirkung
Erste Meldung nach Einrichtung	Zustand wird übernommen, es wird noch nicht geschaltet

6.1 Schutzregeln

Situation	Verhalten
Alarm aktiv, Meldung „aufgeschlossen“	Standort wird unscharf, der Alarm bleibt unquittiert
Alarm aktiv, Meldung „zugeschlossen“	Es wird nicht scharf geschaltet. Nach Quittierung des Alarms holt die Anlage die Scharfschaltung nach, wenn das Gebäude weiterhin als zugeschlossen gemeldet ist
Wartung aktiv	Keine Schaltwirkung, nur Protokoll
Störung der Schnittstelle	Keine Schaltwirkung, nur Protokoll

Der zuletzt gemeldete Gebäudezustand bleibt auch über einen Neustart der Zentrale erhalten.

6.2 Lebenszeichen

Anders als beim Kontakt eines Schlüsselschalters meldet sich ein Ausfall des Drittsystems nicht von selbst – ein Drahtbruch wird erkannt, ein stummes Drittsystem nicht.

Deshalb kann je Zugangsschlüssel ein ****Lebenszeichen**** eingestellt werden: „Meldung erwartet alle ... Minuten“ (Vorschlag 15 Minuten). ****Jede**** Anfrage zählt als Lebenszeichen, auch eine reine Abfrage – ein eigener Aufruf ist nicht nötig.

Einstellung	Verhalten bei ausbleibender Meldung
Lebenszeichen eingestellt	Die Zentrale meldet Störung und friert den Gebäudezustand ein. Der Scharfzustand bleibt unverändert – eine scharfe Anlage wird durch einen Ausfall des Drittsystems nicht unscharf. Mit der nächsten Meldung gilt wieder der gemeldete Zustand.
Kein Lebenszeichen (Voreinstellung)	Der zuletzt gemeldete Zustand gilt unbegrenzt weiter. Ein Ausfall des Drittsystems bleibt unbemerkt. Nur sinnvoll für Systeme, die ausschließlich bei Änderung melden können.

Empfehlung: Das Drittsystem fragt in regelmäßigen Abständen den Zustand ab (Abschnitt 3) und die Zentrale wird mit Lebenszeichen betrieben.

7. Sicherheitsanforderungen

7.1 Übersicht

Anforderung	Beschreibung
Transport	Ausschließlich HTTPS. Anfragen über unverschlüsseltes HTTP werden abgewiesen.

Anforderung	Beschreibung
Schlüsselspeicherung	Auf der Zentrale nur als abgeleiteter Prüfwert, Vergleich in konstanter Zeit
Rechtentrennung	Getrennte Schlüssel für Lesen und für Schreiben
Standortbindung	Ein Schlüssel gilt nur für die ihm zugeordneten Standorte
Absenderbeschränkung	Optional je Schlüssel
Erreichbarkeit	Voreinstellung nur internes Netz, von außen nur auf ausdrücklichen Wunsch
Methoden	Nur GET und POST
Gehäufte Fehlversuche	Ab 10 abgewiesenen Anfragen innerhalb von 10 Minuten meldet die Zentrale eine Störung an den Betreiber (Abschnitt 7.2)
Prüfprotokoll	Jede Anfrage wird auf der Zentrale protokolliert: Zeitpunkt, Absender, Kennung des Schlüssels, Standort, Aktion, Ergebnis. Der Schlüssel selbst wird nie protokolliert.

7.2 Gehäufte Fehlversuche

Die Zentrale **sperrt nicht**, sondern **meldet**: Werden innerhalb von 10 Minuten 10 Anfragen wegen ungültigem Zugangsschlüssel, fehlender Berechtigung oder nicht freigegebenem Standort abgewiesen, setzt sie eine Störung und benachrichtigt den Betreiber mit den Absenderadressen. Die Störung endet 60 Minuten nach dem letzten Fehlversuch.

Bewusst keine Sperre: Eine Sperre würde auch ein falsch eingerichtetes Drittsystem aussperren – oder ein korrektes System, das hinter derselben Internetadresse wie ein Angreifer liegt. Dann kämen auch echte Meldungen wie „aufgeschlossen“ nicht mehr an. Die Meldung macht Angriffe und Fehlkonfiguration sichtbar, ohne den Betrieb zu blockieren. Durchprobieren ist durch die Länge des Zugangsschlüssels und die Begrenzung auf 1 Anfrage pro Sekunde praktisch ausgeschlossen.

Während die Störung besteht, liefert die Abfrage (Abschnitt 3) für alle Standorte **"fault": true**.

7.3 Wiederholerschutz

Wird ein **timestamp** mitgesendet, prüft die Zentrale:

- Er darf höchstens **5 Minuten** von der Uhrzeit der Zentrale abweichen.
- Derselbe Zeitstempel wird je Schlüssel nur **einmal** angenommen.

Damit kann eine mitgeschnittene Anfrage nicht später erneut eingespielt werden. Ohne **timestamp** entfällt dieser Schutz für die betreffende Anfrage; die Uhr des Drittsystems sollte deshalb über NTP synchronisiert sein und der Zeitstempel immer mitgesendet werden.

8. Beispiel: Vollständiger Ablauf

8.1 Schritt 1: Gebäude wird abends zugeschlossen

```
curl -X POST "https://{adresse-der-zentrale}/api/intrusion/access/report" \
-H "Authorization: Bearer {zugangsschluessel}" \
```

```
-H "Content-Type: application/json" \  
-d '{  
  "locationHash": "9f2c4e7a1b3d5f6081a2c3d4e5f60718",  
  "buildingLocked": true,  
  "timestamp": "2026-10-14T18:05:00Z"  
'
```

Response:

```
{ "ok": true, "accepted": { "buildingLocked": true }, "timestamp": "2026-10-14T18:05:00.412Z" }
```

8.2 Schritt 2: Scharfschaltung prüfen

```
curl -X GET "https://{adresse-der-zentrale}/api/intrusion/access/state" \  
-H "Authorization: Bearer {zugangsschluessel}" \  
-H "Accept: application/json"
```

Response während der Vorlaufzeit:

```
{  
  "locations": [  
    { "locationHash": "9f2c4e7a1b3d5f6081a2c3d4e5f60718", "name":  
      "Lagerhalle Nord",  
        "armingState": "arming", "alarmState": "none", "fault": false,  
        "lastReportAt": "2026-10-14T18:05:00.412Z" }  
  ],  
  "timestamp": "2026-10-14T18:05:20.000Z"  
}
```

Nach Ablauf der Vorlaufzeit wechselt **armingState** auf **armed** .

8.3 Schritt 3: Gebäude wird morgens aufgeschlossen

```
curl -X POST "https://{adresse-der-zentrale}/api/intrusion/access/report" \  
-H "Authorization: Bearer {zugangsschluessel}" \  
-H "Content-Type: application/json" \  
-d '{  
  "locationHash": "9f2c4e7a1b3d5f6081a2c3d4e5f60718",
```

```
"buildingLocked": false,
"timestamp": "2026-10-15T06:30:00Z"
}'
```

Der Standort wird sofort unscharf; die Abfrage liefert danach **"armingState": "disarmed"**.

9. Einrichtung

Schritt	Wer	Inhalt
1	Errichter / dexa	Auf der Zentrale eine Zutrittskontroll-Schnittstelle vom Typ „API“ anlegen
2	Errichter / dexa	Zugangsschlüssel erzeugen: Standorte, Recht, optional Lebenszeichen, Absender, Ablauf
3	Errichter / dexa	Adresse, Standort-Kennung(en) und Zugangsschlüssel sicher an den Errichter des Drittsystems übergeben
4	Errichter Drittsystem	Schnittstelle einrichten, Abfrage testen, dann Meldungen „zu“ und „auf“ testen
5	Beide	Ergebnis über die Abfrage und das Prüfprotokoll der Zentrale gemeinsam kontrollieren

Anhang A: Constraints

Feld	Constraint
locationHash	32 Zeichen, hexadezimal (0-9 , a-f), Groß-/Kleinschreibung egal
buildingLocked	JSON-Wahrheitswert true oder false
timestamp	ISO 8601, höchstens 5 Minuten Abweichung zur Zentrale, je Schlüssel nur einmal verwendbar
Größe einer Anfrage	Max. 4 KB
Methoden	GET, POST
Anfragerate	1 pro Sekunde je Absender, kurzzeitig bis zu 5