DEVELOPER DOCUMENTATION

API-Dokumentation

Versionierte, tokenpflichtige Schnittstellen für Fachmodule, Alarmierung, Leitstellenintegration und Lizenzprüfung.

Stand 13.08.2026 · API v1 · Build.B01.0101

AUTH REQUIRED

Authentifizierung und Kontext

Die IGNIS2AB-Fach- und Push-APIs verwenden einen Bearer-Token. Der Token wird für eine konkrete Kombination aus Org-ID und Site-ID erzeugt, nur einmal vollständig angezeigt und danach ausschließlich maskiert gespeichert. Er kann jederzeit widerrufen werden.

```
Authorization: Bearer <site-bound-token>
X-Organization-ID: <org-uuid>
X-Site-ID: <site-uuid>
X-Installation-ID: <installation-uuid>  # optional
Content-Type: application/json
```

Token, Scopes, Benutzerberechtigung, Organisation, Feuerwehr und Objektzugehörigkeit werden gemeinsam geprüft. Tokens gehören niemals in URLs, Browsercode oder Klartextlogs.

Basisadressen

Produktion

`https://app.ignis2ab.com/api/v1`

Integration

`https://dev.ignis2ab.com/api/v1`

Demo

`https://demo.ignis2ab.com/api/v1`

Lizenzdienst

`https://license.2ab.it/api/v1`

Format

JSON · UTF-8 · ISO 8601 · HTTPS

REST API

Fach-API

Organisations- und standortbezogene Daten werden im einheitlichen `data` / `meta`-Format geliefert. Listen sind paginiert.

GET

`/api/v1` – API-Katalog

GET / PATCH

`/organization` – Mandant und sichtbare Feuerwehren

GET / POST

`/personnel` – Personal einer Site lesen oder anlegen

PATCH

`/personnel/{id}` – Person der Site ändern

GET

`/vehicles`, `/equipment`, `/incidents`

GET

`/warehouse/articles`, `/training/bookings`, `/finance/earnings-loss`

GET

`/openapi.json` – laufender API-Vertrag

Scopes

```
api:read  api:docs  organization:read
personnel:read  personnel:write  vehicles:read
equipment:read  incidents:read  warehouse:read
training:read  finance:read
```

HTTP PUSH

AlarmDispatcher- und Leitstellen-API

Die Push-Schnittstelle erzeugt Einsätze aus Alarmierungen und verarbeitet jede Änderung bis Einsatzende. Sie übernimmt Einsatz-ID, Leitstellen-Einsatznummer, Alarmierungs- und Einsatzzeiten, Stichwort, Beschreibung, Ort, Koordinaten, Transportziel, Fahrzeuge, Funkrufnamen, Fahrtnummern, BOS-Status und Personalrückmeldungen.

GET	
`/integrations/alarm-dispatcher/health`	– Token und Site prüfen
POST	
`/integrations/alarm-dispatcher/events`	– Einsatz, Ressourcen und Feedback
POST	
`/integrations/alarm-dispatcher/appointments`	– Termine, Teilnehmende und Rückmeldungen

Rückmeldestatus

```
accepted | declined | maybe | no_response | unknown
```

Idempotenz

Eine stabile `event_id` darf erneut zugestellt werden. Identischer Inhalt wird als Duplikat bestätigt; dieselbe ID mit verändertem Inhalt liefert HTTP 409. Nicht zuordenbare User-IDs werden gezählt und ausgewiesen.

CONNECT API

Herstellerneutrale Alarmierungsanbindung

Für weitere Anbieter können Alarme und Missionen angelegt, aktualisiert, geschlossen oder beendet sowie Feedback und Ressourcen getrennt nachgeliefert werden.

```
GET    /api/v1/connect/health
POST   /api/v1/connect/alarms
```

```
PATCH  /api/v1/connect/alarms/{externalId}
POST   /api/v1/connect/alarms/{externalId}/feedback
POST   /api/v1/connect/alarms/{externalId}/resources
POST   /api/v1/connect/alarms/{externalId}/close
POST   /api/v1/connect/missions
PATCH  /api/v1/connect/missions/{externalId}
DELETE /api/v1/connect/missions/{externalId}
```

Beispiel: Einsatzänderung

```
POST /api/v1/integrations/alarm-dispatcher/events
{
  "event_id": "adc-event-20260813-001",
  "event_type": "mission.updated",
  "occurred_at": "2026-08-13T18:31:22+02:00",
  "mission": {
    "id": "adc-mission-4711",
    "control_center_incident_number": "LS-MSH-2026-004711",
    "status": "active",
    "keyword": "B2 Gebäude",
    "alarmed_at": "2026-08-13T18:27:03+02:00",
    "location":
{"street":"Beispielstraße","house_number":"12","city":"Allstedt"
}
  },
  "vehicles": [{"id":"40-11-1","radio_call_sign":"Florian 40/11-
1","status":"3"}],
  "personnel": [{"user_id":"adc-user-
88421","status":"accepted"}]
}
```

Erfolgsantworten enthalten unter anderem `incident_id`, Anzahl Ressourcen, Feedback und nicht zugeordnete Rückmeldungen.

2AB-License-API

Der eigenständige Lizenzdienst prüft Mandanten- und Site-Lizenzen, bindet Org-ID, Site-ID und Installations-ID und liefert Modus, Laufzeit, Features und Limits. Die Prüfung verwendet Client-ID, Zeitstempel, Nonce und eine HMAC-SHA256-Signatur. Kontrollierte Trial-Provisionierung erfolgt serverseitig per Bearer-Token.

POST

`https://license.2ab.it/api/v1/licenses/verify`

POST

`https://license.2ab.it/api/v1/licenses/trials`

Statuscodes und Betrieb

200 / 202	Verarbeitet oder idempotentes Duplikat
400	Ungültiges JSON
401	Token fehlt, ist ungültig oder abgelaufen
403	Scope, Org-/Site-Zuordnung oder Integration nicht erlaubt
409	Idempotenzkonflikt
415	Content-Type ist nicht JSON

422

Pflichtfeld oder Wert ungültig

Secrets werden getrennt je Umgebung geführt und regelmäßig rotiert. Produktive Integrationen sollten Wiederholungen mit Backoff, stabile externe IDs und eine eigene Zustellprotokollierung verwenden.