GIRA

# Gira IoT REST API – Dokumentation

Stand: 2026-08-31
Version: v3

# Inhaltsverzeichnis



Dieses Dokument beschreibt **Version 3** der Gira IoT REST API. Es ersetzt die Dokumentation zu Version 2 (vom 01.07.2020).

## Neuerungen in Version 3

Im Vergleich zu Version 2:

- Die API-Version `3` ist derzeit die Standardversion; in der Antwort zur Verfügbarkeit wird **`uidLength`** hinzugefügt.
- Die UI-Konfiguration fügt **`iconID`** (Funktionen, Orte, Gewerke) sowie **`scenes`** für jede einzelne Funktion hinzu.
- „Register-Callbacks“ fügt die Option **`batchEvents`** hinzu; Wert-Ereignisse können gebündelt übermittelt werden.
- Wert-Callbacks werden **pro Benutzer gefiltert** (ein Client erhält nur Ereignisse für Funktionen, die sein Benutzer sehen darf) und sind **ratenbegrenzt**.
- Der Endpunkt **Reporting** (`POST /api/messages`) ist nun dokumentiert (er war bereits in Version 2 verfügbar).
- Das Fehlermodell dokumentiert `invalidToken` als **HTTP 401** und fügt **`429`** (`tooManyRequests`) hinzu, das vom Anforderungsratenbegrenzer zurückgegeben wird; `403`/`405` werden nicht ausgegeben.

---

# 1 Über die Gira IoT REST API

Mithilfe der Gira IoT REST API können Sie Datenpunkte unterstützter Geräte programmgesteuert lesen, schreiben und überwachen.

Unterstützte Geräte:

| Gerätetyp | Firmware-Version | Gira IoT REST API-Version |
|---|---|---|
| Gira X1 | 4.0 | v3 |
| Gira One Server | 4.0 | v3 |
| Gira HomeServer / Gira FacilityServer | 4.10 | v2 only |

Der Gira HomeServer / FacilityServer unterstützt ausschließlich die API-Version v2; derzeit ist keine Unterstützung von v3 auf diesen Geräten geplant. Clients, die auf diese Geräte ausgerichtet sind, sollten die Dokumentation zu v2 verwenden.

Funktionen sind die Darstellungen der tatsächlichen Kanäle und Datenpunkte im Rahmen der UI-Konfiguration im Gira Project Assistant (GPA). Die Funktionen sind Teil der UI-Konfiguration, die auch die Orte (Gebäudestruktur) und die Gewerke beschreibt.

Um die API nutzen zu können, muss sich eine Client-Anwendung mit einer Anwendungs-ID registrieren und dabei den Benutzernamen und das Passwort eines Geräts zur Autorisierung verwenden. Die Registrierung für die API erfolgt ausschließlich über HTTPS mit Basis-Zugriffsauthentifizierung.

Um Callbacks nutzen zu können, muss ein separater HTTPS-Server verwendet werden, an den die REST-API die Callback-Ereignisse senden kann.

**Wichtig:** Die HTTPS-Verbindung gilt nicht als vollständig vertrauenswürdig, da es technisch nicht möglich ist, dass der Server ein vertrauenswürdiges TLS-Zertifikat bereitstellt. Aus diesem Grund muss der Client, der für den Zugriff auf die Gira IoT REST API verwendet wird, die Zertifikatsprüfung überspringen. Informationen dazu, wie die Zertifikatsprüfung übersprungen werden kann, finden Sie in der Dokumentation der Client-Software bzw. der Bibliothek.

**Zielgruppe.** Diese Dokumentation richtet sich an Personen, die bereits über folgende Kenntnisse verfügen:

- Grundlegendes Verständnis der HTTP-Kommunikation
- Grundlegendes Verständnis von JSON

## 2 Grundlegende Verwendung

Der Zugriff auf die Gira IoT REST API erfolgt ausschließlich über HTTPS. Alle Daten werden im JSON-Format gesendet und empfangen.

| Anforderungsmethode | Beschreibung |
|---|---|
| `GET` | Ressourcen abrufen. Gibt bei Erfolg den Wert `200 OK` zurück. |
| `PUT` | Eine Ressource aktualisieren. Gibt bei Erfolg den Wert `200 OK` zurück. |
| `POST` | Eine neue Ressource erstellen. Gibt bei Erfolg den Wert `201 Created` zurück. |
| `DELETE` | Eine Ressource löschen. Gibt bei Erfolg den Wert `204 No Content` zurück. |

### Antwortcodes

| Code | Beschreibung |
|---|---|
| `200 OK` | Die Anfrage war erfolgreich, die angeforderte Ressource wird zurückgegeben. |
| `201 Created` | Die Anfrage war erfolgreich, die erstellte Ressource wird zurückgegeben. |
| `204 No Content` | Die Anfrage war erfolgreich, es sind keine Inhalte vorhanden. |
| `400 Bad Request` | Ungültige Anfrage. |
| `401 Unauthorized` | Fehlende oder ungültige Autorisierung. |
| `404 Not Found` | Ressource nicht gefunden. |
| `422 Unprocessable Entity` | Die Anfrage konnte nicht verarbeitet werden. |
| `423 Locked` | Das Gerät ist derzeit gesperrt. |
| `429 Too Many Requests` | In einem bestimmten Zeitraum wurden zu viele Anfragen gesendet. |
| `500 Internal Server Error` | Interner Serverfehler. |

> Hinweis: Unbekannte Ressourcen geben `404 Not Found` (`resourceNotFound`) zurück. Eine Anfrage mit einer nicht unterstützten HTTP-Methode wird von der API nicht beantwortet, und der Front-End-Reverse-Proxy (nginx) gibt `502 Bad Gateway` zurück. Die API gibt weder `403` noch `405` aus; sie gibt `429` (`tooManyRequests`) zurück, wenn eine Quell-IP-Adresse das Limit für fehlgeschlagene Authentifizierungsversuche überschreitet. Die Antworten enthalten den CORS-Header `Access-Control-Allow-Origin: *`.

Im Falle eines Fehlers werden zusätzliche Informationen in einem `error`-Objekt zurückgegeben:

```
{
  "error": {
    "code": "<errorCode>",
    "message": "<errorMessage>"
```

```
  }
}
```

Das Feld `code` enthält einen von mehreren gültigen Fehlercodes, das Feld `message` enthält eine detailliertere Fehlermeldung.

Mögliche Codes:

| Code | HTTP | Nachricht |
|---|---|---|
| `generic` | 500 | … |
| `unprocessable` | 422 | … |
| `invalidAuth` | 401 | Fehlende oder ungültige Authentifizierung. |
| `invalidToken` | 401 | Token fehlt oder ist ungültig. |
| `locked` | 423 | Das Gerät ist derzeit gesperrt. |
| `missingContent` | 400 | Fehlender Inhalt / Fehlender oder ungültiger Wert im Inhalt. |
| `resourceNotFound` | 404 | Ressource … nicht gefunden. |
| `uidNotFound` | 404 | UID … nicht gefunden. |
| `clientNotFound` | 404 | Client … nicht gefunden. |
| `operationNotSupported` | 400 | Die Operation wird für „…“ nicht unterstützt. |
| `callbackTestFailed` | 400 | Der Callback-Test für „…“ ist fehlgeschlagen. |
| `refreshLicensesFailed` | 500 | Lizenzen konnten nicht aktualisiert werden: „…“. |
| `tooManyRequests` | 429 | Zu viele Anfragen. Bitte versuche es später erneut. |
| `unknown` | 500 | Unbekannter Fehler. |

> `invalidAuth` Außerdem enthalten die Antworten von `invalidToken` einen `WWW-Authenticate`-Header (jeweils `Basic realm="REST API"` und `Token realm="REST API"`).

Fehlgeschlagene Authentifizierungsversuche werden pro Quell-IP-Adresse in ihrer Häufigkeit begrenzt: fünf Versuche, die sich innerhalb von fünf Sekunden wieder auffüllen. Für alle Adressen gilt eine globale Obergrenze von zwanzig Versuchen, die sich innerhalb einer Sekunde wieder auffüllen. Bei Überschreitung einer dieser Grenzen wird der Fehler `429 Too Many Requests` (`tooManyRequests`) zurückgegeben, bis das Kontingent wieder aufgefüllt ist; eine erfolgreiche Authentifizierung wird dabei nicht gezählt.

### 2.1 Eindeutige Kennung

Die API basiert auf dem Konzept der eindeutigen Identifikatoren (UID), die kurz (vier Zeichen), dauerhaft und ohne sofortige Wiederholung sind.

Die vier Zeichen lange UID besteht aus Kleinbuchstaben und Ziffern und beginnt mit einem Buchstaben. Die Länge der UID wird bei der Verfügbarkeitsprüfung ebenfalls als `uidLength` angegeben (siehe Abschnitt 3).

### 2.2 Versionsverwaltung über den URL-Pfad

Die Gira IoT REST API unterstützt den Zugriff auf verschiedene Versionen der API über den URL-Pfad in den Anfragen. Die verfügbaren Versionen sind `v1`, `v2` und `v3`.

```
GET /api/v3/resource
```

Wenn der Versionsteil in der URL weggelassen wird, wird die neueste verfügbare Version (derzeit **v3**) verwendet. Bei der Anforderung einer Version außerhalb des unterstützten Bereichs wird `404 Not Found` zurückgegeben.

### 2.3 Genehmigung

Für alle API-Anfragen mit Ausnahme der API-Verfügbarkeitsprüfung ist eine Registrierung des Clients mit einer Autorisierung in Form von Benutzername und Passwort erforderlich.

Um einen Client zu registrieren, müssen die Anmeldedaten als HTTP-Basic-Authentifizierung im Header `Authorization` angegeben werden. Dadurch wird auf dem Server ein neuer Client angelegt und ein eindeutiges Token zurückgegeben. Für nachfolgende API-Aufrufe ist dieses Token zur Autorisierung erforderlich; es muss **entweder** als Bearer-Token im Header `Authorization` **oder** als URL-Abfrageparameter angegeben werden:

```
GET /api/v3/resource
Authorization: Bearer <token>
```

```
GET /api/v3/resource?token=<token>
```

Sind beide vorhanden, hat der Header `Authorization` Vorrang.

## 3 Prüfung der API-Verfügbarkeit

Die Verfügbarkeit der Gira IoT REST API lässt sich durch Aufrufen der Basis-URL überprüfen.

```
GET /api/v3/
```

**Antwort**

```
{
  "info": "GDS-REST-API",
  "version": "3",
  "deviceName": "<display name>",
  "deviceType": "<device type identifier>",
  "deviceVersion": "<device version number>",
  "uidLength": 4
}
```

| Feld | Beschreibung |
|---|---|
| `info` | Immer die Zeichenfolge `GDS-REST-API`. |
| `version` | Die Versionsnummer der API (als Zeichenfolge). |
| `deviceName` | Benutzerdefinierter Anzeigename des Geräts. Wird von v2 zurückgegeben. |
| `deviceType` | Gerätetyp-Kennung, zum Beispiel `GIGSRVKX02` für Gira X1. Wird von v2 zurückgegeben. |
| `deviceVersion` | Firmware-Version des Geräts im Format `n.n.n.n`, zum Beispiel `4.0.348.0`. Zurückgegeben von v2. |
| `uidLength` | Länge einer UID-Zeichenkette. Derzeit `4`. **Neu in Version 3.** |

Dieser Endpunkt erfordert keine Autorisierung und wird auch bei gesperrtem Gerät nicht blockiert.

# 4 Anmeldung

## 4.1 Client-ID

Jeder Client muss eine eindeutige Client-Kennung verwenden. Um die Eindeutigkeit zu gewährleisten, müssen Client-Kennungen URNs innerhalb der Organisation des Clients sein (z. B. `de.gira.gdsrestapi.clients.my_well_known_service`).

## 4.2 Client registrieren

Eine clientseitige Registrierung bei der API ist erforderlich, um die Identität des Clients sowie optionale Callback-URLs für die Ereignisbenachrichtigung zu verwalten. Falls der Client über Konfigurationsänderungen oder Wertänderungen benachrichtigt werden möchte, können nach der Registrierung entsprechende Callback-URLs angegeben werden. Es werden ausschließlich HTTPS-basierte Callback-URLs unterstützt.

Die API gibt bei erfolgreicher Registrierung ein Token für den Client-Zugriff zurück, das bei allen nachfolgenden API-Aufrufen zur Authentifizierung verwendet werden muss.

```
POST /api/clients
```

```
{ "client": "<client identifier>" }
```

### Antwort

```
{ "token": "<client access token>" }
```

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 201 Created | | Der Client wurde erfolgreich registriert. |
| 400 Bad Request | `missingContent` | Der Textkörper ist leer/ungültig. |
| 401 Unauthorized | `invalidAuth` | Fehlende oder ungültige Authentifizierung. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |

Das Token ist eine zufällig generierte Zeichenfolge, die aus 32 alphanumerischen Zeichen (Groß- und Kleinbuchstaben) besteht (Regex: `[0-9A-Za-z]{32}`).

Jedes Mal, wenn eine neue Kombination aus Benutzername aus dem HTTP-Basic-Authorization-Header und der `<client-identifier>` verwendet wird, wird ein neues Token generiert. Wenn man sich erneut mit demselben Benutzernamen und derselben `<client-identifier>` registriert, ohne das Token zuvor zu deaktivieren, wird dasselbe Token generiert. Es können mehrere Clients gleichzeitig registriert werden, und die Gültigkeitsdauer eines Tokens ist nicht begrenzt.

Das Token bleibt gültig, bis der Client abgemeldet, der Benutzer gelöscht oder der Benutzername geändert wird. Das Token bleibt gültig, wenn das Passwort des Benutzers geändert wird. Ein Token, das aufgrund eines gelöschten Benutzernamens ungültig geworden ist, wird wieder gültig, sobald der Benutzername erneut verwendet wird.

### 4.3 Client abmelden

```
DELETE /api/clients/<access token>
```

#### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 204 No Content | | Der Client wurde erfolgreich abgemeldet. |
| 401 Unauthorized | `invalidToken` | Das Token wurde nicht gefunden. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `generic` | Der Client konnte nicht entfernt werden. |

### 4.4 Callbacks registrieren

```
POST /api/clients/<access token>/callbacks
```

```
{
  "serviceCallback": "<callback URL of service events, optional>",
  "valueCallback": "<callback URL of value events, optional>",
  "testCallbacks": "<boolean value, optional>",
  "batchEvents": "<boolean value, optional>"
}
```

| Feld | Beschreibung |
|---|---|
| `serviceCallback` | HTTPS-URL, die Service-Ereignisse empfängt. Optional. |
| `valueCallback` | HTTPS-URL, an die Wert-Ereignisse gesendet werden. Optional. |
| `testCallbacks` | Wenn `true` gesetzt ist, wird jede angegebene Callback-URL bei der Registrierung abgefragt und muss mit `200 OK` antworten. Optional, Standardwert: `false`. |
| `batchEvents` | **Neu in Version 3.** Wenn `true` gesetzt ist, werden Ereignisdaten für diesen Client gebündelt übermittelt (mehrere Ereignisse pro Callback-Anfrage) statt eines Ereignisses pro Anfrage. Optional, Standardwert ist `false`. Siehe Abschnitt 4.7 „“. |

Mindestens eine der folgenden Adressen muss angegeben werden: `serviceCallback` oder `valueCallback`. Pro `<access token>` kann nur ein Satz von Callback-URLs registriert werden; eine erneute Registrierung ersetzt den bisherigen Satz.

**Anforderungen an Callback-URLs.** Jede Callback-URL muss eine absolute `https://`-URL in der Form `https://<host>[:<port>]/<path>[?<query>]` sein. Der Host muss vom Gerät aus erreichbar sein und darf keine Loopback- oder Localhost-Adresse sein. URLs, die kein HTTPS verwenden, Steuerzeichen (einschließlich CR/LF) enthalten oder anderweitig fehlerhaft sind, werden mit dem Fehler `422 unprocessable` abgelehnt.

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Der Callback wurde erfolgreich registriert. |
| 400 Bad Request | `missingContent` | Der Textkörper ist leer/ungültig. |
| 400 Bad Request | `callbackTestFailed` | Der Callback-Test hat nicht mit `200 OK` geantwortet. |
| 401 Unauthorized | `invalidToken` | Das Token wurde nicht gefunden. |
| 422 Unprocessable | `unprocessable` | Die Callback-URL ist ungültig – sie lautet nicht `https://`, enthält Steuerzeichen, verweist auf „localhost“/„loopback“ oder ist fehlerhaft. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |

## 4.5 Callbacks entfernen

```
DELETE /api/clients/<access token>/callbacks
```

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Der Callback wurde erfolgreich entfernt. |
| 401 Unauthorized | `invalidToken` | Das Token wurde nicht gefunden. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `generic` | Der Callback konnte nicht entfernt werden. |

## 4.6 Service-Callback

```
POST <callback URL of service events>
```

### Hauptteil

```
{
  "token": "<client access token>",
  "events": [ { "event": "<event type>" }, ... ],
  "failures": "<number of unsuccessful callback attempts since last successful one,
optional>"
}
```

### Antwortcodes

- `200 OK`
- `404 Not Found` — Wenn der Client auf das Callback-Ereignis mit `404 Not Found` reagiert, hebt die API die Registrierung des Clients implizit auf.

### Veranstaltungstypen

| Veranstaltung | Beschreibung |
|---|---|
| `test` | Der Ereignistyp „Callback-Test“. Wird ausschließlich im Szenario „Registrieren und Testen“ verwendet. |
| `startup` | Das Gerät ist betriebsbereit. |
| `restart` | Das Gerät wird neu gestartet. Gesendet mit einem `Connection: close`-Header. |
| `projectConfigChanged` | Die Projektkonfiguration hat sich geändert (z. B. GPA-Download). In diesem Fall hat sich die UI-Konfiguration möglicherweise nicht ebenfalls geändert. Es wird nicht empfohlen, sofort auf dieses Ereignis zu reagieren, da der REST-Server bei Auslösung dieses Ereignisses noch einige Sekunden lang gesperrt bleibt. |
| `uiConfigChanged` | Die UI-Konfiguration hat sich geändert. |

## 4.7 Wert-Callback

```
POST <callback URL of value events>
```

### Hauptteil

```
{
  "token": "<client access token>",
  "events": [ { "uid": "<unique identifier of data point>", "value": "<new data point
value>" }, ... ],
  "failures": "<number of unsuccessful callback attempts since last successful one,
optional>"
}
```

Bei einem Callback-Test ist `events` eine leere Liste.

### Antwortcodes

- `200 OK`
- `404 Not Found` — Wenn der Client auf das Callback-Ereignis mit `404 Not Found` reagiert, hebt die API die Registrierung des Clients implizit auf.

### Übertragung, Filterung und Ratenbegrenzung

- **Filterung pro Benutzer:** Ein Client erhält nur Wert-Ereignisse für Datenpunkte, die zu Funktionen gehören, für die sein Benutzer eine Zugriffsberechtigung hat. Ereignisse für andere Datenpunkte werden nicht übermittelt.
- **Batching:** Wenn sich der Client bei `batchEvents: true` registriert hat (Abschnitt 4.4), werden Wert-Ereignisse gesammelt und als Array übermittelt; andernfalls wird jedes Ereignis in einer eigenen Anfrage übermittelt.

- **Ratenbegrenzung:** Um Clients – sowie Dienste, die Ereignisse von vielen Geräten aggregieren – vor Ereignisstürmen zu schützen, werden Wert-Ereignisse gedrosselt. Wenn ein einzelner Datenpunkt innerhalb von 5 Sekunden 50 Ereignisse überschreitet, werden weitere Ereignisse für diesen Datenpunkt bis zur nächsten Änderung der Projektkonfiguration unterdrückt. Treten innerhalb von 5 Sekunden über alle Datenpunkte hinweg mehr als 500 Wert-Ereignisse auf, wird eine globale Schutzmaßnahme ausgelöst und **alle** Wert-Ereignisse werden **bis zum Neustart des Geräts** unterdrückt – dies ist beabsichtigt und wird bei einer Konfigurationsänderung nicht zurückgesetzt. Entwerfen Sie Clients so, dass sie keine solchen Spitzen erzeugen. Wertaktualisierungen mit fehlendem oder fehlerhaftem Status werden nicht übermittelt.

# 5 UI-Konfiguration

## 5.1 UI-Konfiguration-Kennung

```
GET /api/uiconfig/uid
```

### Antwort

Eindeutige Kennung der aktuellen Konfiguration. Diese Kennung ändert sich bei jeder Änderung der Konfiguration (z. B. beim Herunterladen eines GPA-Projekts oder bei Konfigurationsänderungen über die Gira Smart Home App).

```
{ "uid": "<unique ui configuration identifier>" }
```

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Die aktuelle Konfigurationskennung wird zurückgegeben. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `unknown` | Das Gerät konnte die Konfigurations-ID nicht bereitstellen. |

## 5.2 Konfiguration abrufen

```
GET /api/uiconfig[?expand=dataPointFlags,parameters,locations,trades]
```

### Antwort

Laden Sie die vollständige UI-Konfiguration herunter.

- `uid` — Die eindeutige Kennung für die UI-Konfiguration.
- `functions` — Eine Liste aller Funktionen.
  - `uid` — Die eindeutige Kennung der Funktion.
  - `functionType` — Der eindeutige Ressourcenname des Funktionstyps. Siehe Abschnitt 10.1, Funktionsdefinitionen.
  - `channelType` — Eindeutiger Ressourcenname des Kanaltyps. Siehe Abschnitt 10.2, Kanaldefinitionen.
  - `displayName` — UTF-8-basierter Anzeigename.
  - `iconID` — Numerische Symbol-Kennung der Funktion (falls nicht vorhanden: `-1`). **Neu in Version 3.**
  - `dataPoints` — Eine Liste aller in der Funktion verfügbaren Datenpunkte.
    - `uid` — Die eindeutige Kennung des Datenpunkts.
    - `name` — Der logische Name des Datenpunkts basierend auf der Kanaldefinition.

  - `canRead` / `canWrite` / `canEvent` — Funktionen für Datenpunkte. Wird nur zurückgegeben, wenn `dataPointFlags` im Parameter `expand` enthalten ist.
  - `parameters` — Eine Liste von Funktionsparametern. Wird nur zurückgegeben, wenn sie im Parameter `expand` enthalten ist.
  - `scenes` — Szenendefinitionen der Funktion, jeweils ein Objekt vom Typ `{ "name": <string>, "number": <int> }`. **Neu in Version 3.** Wird nur zurückgegeben, wenn die Funktion Szenen enthält.
- `locations` — Eine verschachtelte Liste aller Orte und der darin enthaltenen eindeutigen Funktionskennungen. Wird nur zurückgegeben, wenn diese im Parameter `expand` vorhanden sind. Jeder Ort verfügt über `displayName`, `locationType`, `iconID` und `functions` und kann verschachtelte `locations` enthalten.
- `trades` — Eine Liste aller Gewerke und der darin enthaltenen eindeutigen Funktionskennungen. Wird nur zurückgegeben, wenn sie im Parameter `expand` vorhanden sind. Jeder Trade verfügt über eine `displayName`, `tradeType`, `iconID` und `functions`.

```
{
  "uid": "<unique ui configuration identifier>",
  "functions": [
    {
      "uid": "<unique function id>",
      "functionType": "<function type urn>",
      "channelType": "<channel type urn>",
      "displayName": "<display name>",
      "iconID": -1,
      "dataPoints": [
        {
          "uid": "<unique id>",
          "name": "<name within channel definition>",
          "canRead": true,
          "canWrite": true,
          "canEvent": true
        }
      ],
      "parameters": [
        { "set": "<set name>", "key": "<key name>", "value": "<value>" }
      ],
      "scenes": [ { "name": "All on", "number": 2 } ]
    }
  ],
  "locations": [
    {
      "displayName": "<display name>",
      "locationType": "<predefined location type>",
      "iconID": -1,
      "functions": [ { "uid": "<unique function identifier>" } ],
      "locations": [ "..." ]
    }
  ],
  "trades": [
    {
      "displayName": "<display name>",
      "tradeType": "<predefined trade type>",
      "iconID": -1,
      "functions": [ { "uid": "<unique function identifier>" } ]
```

```
    }
  ]
}
```

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Die UI-Konfiguration wird zurückgegeben. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 401 Unauthorized | `invalidAuth` | Das Gerät hat die Anfrage für diesen Benutzer abgelehnt. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `unknown` | Das Gerät konnte die Konfiguration nicht bereitstellen. |

# 6 Werte

Die Werte der Datenpunkte werden als JSON ausgetauscht. Beim Auslesen von Werten (Abschnitt 6.1) gibt die API jeden `value` als **JSON-String** zurück (zum Beispiel `"20"` oder `"70.196078"`), unabhängig vom zugrunde liegenden Datentyp. Beim Schreiben von Werten (Abschnitt 6.2, Abschnitt 6.3) kann das `value` als JSON-Zeichenkette oder als JSON-Primitiv angegeben werden (beispielsweise `20` oder `true`).

## 6.1 Wert(e) abrufen

```
GET /api/values/<uid>
```

Die UID kann Folgendes bezeichnen:

- Ein Datenpunkt; in diesem Fall wird nur der Wert dieses Datenpunkts zurückgegeben.
- Eine Funktion, bei der in diesem Fall alle Datenpunktwerte der Funktion zurückgegeben werden.

### Antwort

Der tatsächliche Wert des/der referenzierten Datenpunkte(s).

```
{
  "values": [
    { "uid": "<unique data point identifier>", "value": "<actual value>" }
  ]
}
```

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Zurückgegebene Werte. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 401 Unauthorized | `invalidAuth` | Die UID ist für den Benutzer nicht sichtbar – dies gilt auch für unbekannte oder fehlerhafte UIDs, da zunächst die Sichtbarkeit geprüft wird. |
| 404 Not Found | `uidNotFound` | Sonderfall: Die UID ist sichtbar, weist jedoch keine Zuordnung zu einem Datenpunkt auf. |
| 422 Unprocessable | `unprocessable` | Die UID bezieht sich nicht auf einen lesbaren Datenpunkt. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `generic` | Das Gerät konnte den Wert nicht bereitstellen. |

### 6.2 Wert(e) festlegen

```
PUT /api/values
```

**Anfrage**

Der Wert bzw. die Werte der angegebenen Datenpunkte.

```
{
  "values": [
    {
      "uid": "<unique data point identifier>",
      "value": "<actual value>",
      "hint": "<field bus specific additional information, optional>"
    }
  ]
}
```

Datenpunkte, die für den Benutzer nicht sichtbar sind, werden ohne Rückmeldung übersprungen. Sind keine einstellbaren Datenpunkte mehr vorhanden, wird `422 unprocessable` zurückgegeben.

**Antwortcodes**

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Die Werte wurden festgelegt. |
| 400 Bad Request | `missingContent` | `values` fehlt oder ist kein Array. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 401 Unauthorized | `invalidAuth` | Das Gerät hat die Anfrage für diesen Benutzer abgelehnt. |
| 422 Unprocessable | `unprocessable` | Ein Eintrag ist ungültig, es sind keine einstellbaren Werte mehr vorhanden oder das Gerät konnte einen oder mehrere Werte nicht einstellen. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |

### 6.3 Einzelwert festlegen

```
PUT /api/values/<uid>
```

**Anfrage**

Der Wert des angegebenen Datenpunkts.

```
{ "value": "<actual value>" }
```

**Antwortcodes**

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Der Wert wurde festgelegt. |
| 400 Bad Request | `missingContent` | `value` fehlt oder ist keine Primitive. |
| 400 Bad Request | `operationNotSupported` | Die UID bezieht sich nicht auf einen beschreibbaren Datenpunkt. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 401 Unauthorized | `invalidAuth` | Die UID ist für den Benutzer nicht sichtbar – sie wird auch bei unbekannten oder fehlerhaften UIDs zurückgegeben. |
| 404 Not Found | `uidNotFound` | Sonderfall: Die UID ist sichtbar, weist jedoch keine Zuordnung zu einem Datenpunkt auf. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `generic` | Das Gerät konnte den Wert nicht einstellen. |

# 7 Berichterstattung

Eine generische Melde-Schnittstelle, über die Clients Fehler, Warnungen und allgemeine Meldungen melden können, damit diese in das Gerätelogbuch geschrieben werden.

## 7.1 Eine Nachricht an das Gerät senden

```
POST /api/messages
```

### Anfrage

```
{
  "messageType": "<syslog compliant severity level>",
  "message": "<message text>",
  "timestamp": "<ISO 8601 timestamp>",
  "payload": "<payload object, optional>"
}
```

- `messageType` — eine syslog-konforme Schweregradstufe.
- `message` — eine Nachricht auf Englisch.
- `timestamp` — eine Darstellung von Datum und Uhrzeit gemäß ISO 8601 (z. B. `2012-04-23T18:25:43.511Z`).
- `payload` — ein beliebiges JSON-Objekt mit weiteren Informationen zur Nachricht.

Der Hauptteil muss gültiges JSON sein. In der aktuellen Implementierung wird die oben genannte Struktur nicht erzwungen – der Hauptteil wird zusammen mit dem Benutzer und der Quell-IP-Adresse in das Geräteprotokoll geschrieben.

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Nachricht angenommen. |
| 400 Bad Request | `missingContent` | Der Textkörper ist leer/ungültig. |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |

# 8 Lizenzen

## 8.1 Lizenzen erwerben

```
GET /api/licenses[?refresh=true]
```

Ruft alle auf dem Gerät verfügbaren Lizenzen ab. Wenn der optionale Parameter `refresh=true` angegeben wird, aktualisiert das Gerät zunächst die Lizenzen vom Lizenzserver. Erfordert API-Version v2 oder höher.

### Antwort

Die Liste aller verfügbaren Lizenzen.

```
{
  "licenses": [
    {
      "vendor": "<vendor>",
      "domain": "<domain>",
      "feature": "<feature>",
      "name": { "<ISO 639-1 language code>": "<translated name>" },
      "uuid": "<uuid>",
      "creationDate": "<creation date>",
      "validTo": "<optional expiry date>",
      "target": "<optional target>",
      "configurationHint": "<optional configuration hint>"
    }
  ]
}
```

| Feld | Beschreibung |
|---|---|
| `vendor` | Der Anbieter der lizenzierten Funktion. |
| `domain` | Der Bereich der lizenzierten Funktion. |
| `feature` | Die lizenzierte Funktion. |
| `name` | Der optionale Name der Lizenz als Liste übersetzter Einträge mit einem ISO 639-1-Sprachcode als Schlüssel. |
| `uuid` | Die UUID der Lizenz. |
| `creationDate` | Das Erstellungsdatum der Lizenz im ISO-8601-Format `YYYY-MM-DDThh:mm:ssZ`, zum Beispiel `2019-05-22T11:16:46Z`. |
| `validTo` | Das optionale Ablaufdatum der Lizenz im ISO 8601-Format `YYYY-MM-DD`, zum Beispiel `2025-12-31`. |
| `target` | Das optionale Ziel für die Lizenz. |
| `configurationHint` | Die optionalen spezifischen Konfigurationsinformationen für die Funktion. |

### Antwortcodes

| HTTP-Code | Fehlercode | Beschreibung |
|---|---|---|
| 200 OK | | Zurückgegebene Lizenzen (leere Liste, falls keine vorhanden). |
| 401 Unauthorized | `invalidToken` | Token fehlt oder ist ungültig. |
| 423 Locked | `locked` | Das Gerät ist derzeit gesperrt. |
| 429 Too Many Requests | `tooManyRequests` | Die Quell-IP-Adresse wird nach zu vielen fehlgeschlagenen Authentifizierungsversuchen vorübergehend gesperrt. |
| 500 Internal Server Error | `refreshLicensesFailed` | `refresh=true` wurde angefordert, und die Aktualisierung ist fehlgeschlagen oder es ist eine Zeitüberschreitung aufgetreten. |

# 9 Beispiele

Beispiel für ein GPA-Projekt eines X1 mit zwei Dimmerfunktionen und einer Sonos-Audiofunktion. IP-Adresse des X1: `192.168.137.173`. Anmeldedaten des Geräts: Benutzername `device`, Passwort `device`.

## 9.1 Registrierung eines Clients

### Anfrage

```
POST /api/v3/clients HTTP/1.1
Host: 192.168.137.173
Content-Type: application/json
Authorization: Basic dXNlcjpwYXNzd29yZA==

{"client":"de.example.myapp"}
```

### Antwort

```
{ "token": "wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

## 9.2 Prüfung der API-Verfügbarkeit

### Anfrage

```
GET /api/v3/ HTTP/1.1
Host: 192.168.137.173
```

### Antwort

```
{
  "info": "GDS-REST-API",
  "version": "3",
  "deviceName": "My X1",
  "deviceType": "GIGSRVKX02",
  "deviceVersion": "4.0.348.0",
  "uidLength": 4
}
```

## 9.3 Abrufen der UI-Konfiguration

### Anfrage (ohne Erweiterungsparameter)

```
GET /api/v3/uiconfig?token=wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD HTTP/1.1
Host: 192.168.137.173
```

### Verkürzte Antwort

```
{
  "functions": [
    {
      "channelType": "de.gira.schema.channels.KNX.Dimmer",
      "dataPoints": [
        { "name": "OnOff", "uid": "a02a" },
        { "name": "Brightness", "uid": "a02b" }
      ],
```

```
      "displayName": "Lampe Links",
      "functionType": "de.gira.schema.functions.KNX.Light",
      "iconID": -1,
      "uid": "a029"
    }
  ],
  "uid": "a036"
}
```

### 9.4 Werte festlegen

**Helligkeit der linken Lampe auf 70 % einstellen**

```
PUT /api/v3/values/a02b?token=wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD HTTP/1.1
Host: 192.168.137.173
Content-Type: application/json

{"value":70}
```

**Helligkeit beider Lampen auf 20 %/30 % einstellen**

```
PUT /api/v3/values?token=wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD HTTP/1.1
Host: 192.168.137.173
Content-Type: application/json

{ "values": [ { "uid": "a02b", "value": 20 }, { "uid": "a02e", "value": 30 } ] }
```

### 9.5 Werte abrufen

**Anfrage**

```
GET /api/v3/values/a02b?token=wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD HTTP/1.1
Host: 192.168.137.173
```

**Antwort**

```
{ "values": [ { "uid": "a02b", "value": "20" } ] }
```

### 9.6 Lizenzen erwerben

**Anfrage**

```
GET /api/v3/licenses HTTP/1.1
Host: 192.168.137.173
Authorization: Bearer wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD
```

**Antwort**

```
{
  "licenses": [
    {
      "creationDate": "2019-11-04T11:48:47Z",
      "domain": "test_licenses",
      "feature": "new_test_license",
```

```
      "name": { "de": "neue lizenz", "en": "new license" },
      "uuid": "5cd92c49-8686-4c54-8b9a-b6eb62f35795",
      "vendor": "gira_de"
    }
  ]
}
```

### 9.7 Callbacks

Callback-Server unter `192.168.137.128:5523`, der Service-Callbacks unter `/service` und Wert-Callbacks unter `/value` empfängt.

#### Callbacks registrieren

```
POST /api/v3/clients/wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD/callbacks HTTP/1.1
Host: 192.168.137.173
Content-Type: application/json

{
  "serviceCallback": "https://192.168.137.128:5523/service",
  "valueCallback": "https://192.168.137.128:5523/value",
  "testCallbacks": true
}
```

#### Callback-Test auf dem Callback-Server empfangen

```
POST /service HTTP/1.1
Host: 192.168.137.128:5523
Content-Type: application/json

{ "events": [ { "event": "test" } ], "token": "wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

```
POST /value HTTP/1.1
Host: 192.168.137.128:5523
Content-Type: application/json

{ "events": [], "token": "wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

#### Callback-Wert, wenn die linke Lampe auf 70 % eingestellt ist

```
POST /value HTTP/1.1
Host: 192.168.137.128:5523
Content-Type: application/json

{ "events": [ { "uid": "a02b", "value": "70" } ], "failures": 0, "token":
"wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

#### Service-Callbacks beim Neustart (mit einem zuvor verpassten Ereignis)

```
POST /service HTTP/1.1
Host: 192.168.137.128:5523
Connection: close
Content-Type: application/json
```

```
{ "events": [ { "event": "restart" } ], "failures": 1, "token":
"wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

```
POST /service HTTP/1.1
Host: 192.168.137.128:5523
Content-Type: application/json

{ "events": [ { "event": "startup" } ], "failures": 0, "token":
"wlkTNYIsYLJLsrw68Z3goYdQf6uiO4jD" }
```

# 10 Unterstützte Funktionen

Alle im GPA verfügbaren Funktionen werden auch über die Gira IoT REST API unterstützt. Funktionen, die in neueren Firmware-Versionen des Gira-Servergeräts hinzugefügt werden, stehen automatisch in der Gira IoT REST API zur Verfügung, auch wenn sich die Version der API nicht ändert.

Die folgenden Tabellen wurden anhand der in der Geräte-Firmware (**Gira X1 4.0** und **Gira One Server 4.0**) enthaltenen Definitionen von Funktionen, Kanälen und Datenpunkten erstellt und geben wieder, welche Funktionen die einzelnen Geräte tatsächlich bereitstellen. HomeServer / FacilityServer unterstützen ausschließlich API v2 – Informationen zum Funktionsumfang finden Sie in der Dokumentation zu v2.

Die Kanaltypen `Switch2`, `Blind2`, `DimmerRGBW2` und `Trigger2` sowie der Funktionstyp `de.gira.schema.functions.Camera2` wurden mit Gira X1 4.0 und Gira One Server 4.0 eingeführt. Ein Gerät mit einer älteren Firmware stellt diese nicht zur Verfügung; ein Client, der diese benötigt, sollte die vom Verfügbarkeitscheck gemeldete Geräteversion überprüfen (siehe Abschnitt 3).

Ein am Ende stehendes `2` ist kein allgemeiner Hinweis darauf. In den Namen von Datenpunkten handelt es sich dabei um einen Instanzindex und nicht um eine Version: Der Kanal `GiraOne.WeatherStationVisu` umfasst die Werte von `Sun-Protection-1` bis `Sun-Protection-4`.

## 10.1 Funktionsdefinitionen

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| Switched light | de.gira.schema.functions.Switch | Switch, Switch2 | Schalte einen Datenpunkt zwischen 0 und 1 um. | X | X |
| Dimmer | de.gira.schema.functions.KNX.Light | KNX.Dimmer | Steuern Sie einen Dimmer mit optionaler Verschiebung und/ oder Helligkeit. | X | X |
| Colored light | de.gira.schema.functions.ColoredLight | DimmerRGBW, DimmerRGBW2 | Eine RGB(W)-Farbe mit optionaler Helligkeit festlegen. | X | X |
| Tunable white | de.gira.schema.functions.TunableLight | DimmerWhite | Stellen Sie eine Farbtemperatur mit optionaler Helligkeit ein. | X | X |
| Philips Hue light | de.gira.schema.functions.Hue.Light | Hue.Light | Eine Philips Hue-Lampe steuern. | X | X |
| Shutter and blind | de.gira.schema.functions.Covering | BlindWithPos, Blind2 | Rollladen/Jalousie bewegen; optional absolute Position und/ oder Lamellenposition. | X | X |

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| Heating and cooling | de.gira.schema.functions.KNX.HeatingCooling | KNX.HeatingCoolingSwitchable | Heizung/Kühlung steuern. | X | X |
| Heating and cooling (relative) | de.gira.schema.functions.KNX.HeatingCoolingRelative | KNX.HeatingCoolingRelative | Regeln Sie die Heizung/Kühlung mit einem relativen Sollwert. | X | |
| KNX fan coil / air conditioning | de.gira.schema.functions.KNX.FanCoil | KNX.FanCoil | Steuerung der Klimaanlage / des Gebläsekonvektors. | X | |
| Sauna temperature | de.gira.schema.functions.SaunaHeating | RoomTemperatureSwitchable | Stellen Sie die Saunatemperatur ein, wahlweise mit Ein-/Aus-Funktion. | X | |
| Weather station | de.gira.schema.functions.KNX.WeatherInformation | KNX.WeatherInformation | Wetterstations- und Klimawerte anzeigen. | X | X |
| Threshold setting | de.gira.schema.functions.ThresholdSetting | ThresholdSetting | Legen Sie einen Schwellenwert bzw. Grenzwert fest. | X | X |
| Trigger on/off | de.gira.schema.functions.Trigger | Trigger, Trigger2 | Setzen Sie einen Datenpunkt | X | X |

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| | | | auf den vordefinierten Wert 0 oder 1. | | |
| Press and hold | de.gira.schema.functions.PressAndHold | Trigger, Trigger2 | Stellen Sie beim Drücken auf 0/1 und beim Loslassen auf 1/0 ein. | X | |
| IoT trigger | de.gira.schema.functions.iot.Trigger | iot.Trigger | Eine IoT-/Automatisierungsaktion auslösen. | X | |
| Scene | de.gira.schema.functions.FunctionScene | FunctionScene | Die Szenen einer Funktion ausführen bzw. einüben. | X | X |
| Scene (legacy) | de.gira.schema.functions.Scene | SceneSet, SceneControl | Veraltete Szene „Ausführen/Lernen“ (ersetzt durch „Funktionsszene“). | X | |
| Audio | de.gira.schema.functions.Audio | AudioWithPlaylist, AudioWithCover | Einen Audio-Player steuern. | X | X |
| Sonos audio | de.gira.schema.functions.Sonos.Audio | Sonos.Audio | Einen Sonos-Audio-Player steuern. | X | X |

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| IP camera | de.gira.schema.functions.Camera2 | Camera | Eine IP-Kamera anzeigen. | X | X |
| ONVIF camera | de.gira.schema.functions.Onvif.Camera | Onvif.Camera | Zeigen Sie eine ONVIF-konforme IP-Kamera an. | X | X |
| IP camera (legacy) | de.gira.schema.functions.Camera | Camera | Ältere IP-Kamera (durch IP-Kamera abgelöst). | X | |
| Link | de.gira.schema.functions.LinkOverlay | Link | Eine Website bzw. einen Link als Overlay anzeigen. | X | X |
| Link (legacy) | de.gira.schema.functions.Link | Link | Link zur alten Website (ersetzt durch Link). | X | |
| Binary status value | de.gira.schema.functions.BinaryStatus | Binary | Zeige einen Binärwert an. | X | X |
| Unsigned status value | de.gira.schema.functions.NumericUnsignedStatus | DWord | Zeige einen vorzeichenlosen Wert an. | X | |
| Signed status value | de.gira.schema.functions.NumericSignedStatus | Integer | Zeige einen vorzeichenbehafteten Wert an. | X | |
| Decimal status value | de.gira.schema.functions.NumericFloatStatus | Float | Zeige einen Dezimalwert an. | X | X |

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| Text status value | de.gira.schema.functions.TextStatus | String | Einen Textwert anzeigen. | X | |
| Multi-status | de.gira.schema.functions.multi.Status | multi.Status | Einen kombinierten Wert mit mehreren Status anzeigen. | X | |
| Unsigned value transmitter (8-bit) | de.gira.schema.functions.Unsigned8BitValue | Byte | Einen vorzeichenlosen 8-Bit-Wert festlegen. | X | |
| Signed value transmitter (8-bit) | de.gira.schema.functions.Signed8BitValue | Integer | Einen vorzeichenbehafteten 8-Bit-Wert festlegen. | X | |
| Percent value transmitter | de.gira.schema.functions.PercentValue | Percent | Geben Sie einen Prozentsatz ein. | X | |
| Temperature value transmitter | de.gira.schema.functions.TemperatureValue | Temperature | Legen Sie einen Temperaturwert fest. | X | |
| Unsigned value transmitter | de.gira.schema.functions.UnsignedValue | DWord | Einen vorzeichenlosen Wert festlegen. | X | |
| Signed value transmitter | de.gira.schema.functions.SignedValue | Integer | Einen vorzeichenbehafteten Wert festlegen. | X | |

| Funktionsname | Funktions-URN | Kanaltyp(en) | Beschreibung | X1 | Gira One |
|---|---|---|---|---|---|
| Decimal value transmitter | de.gira.schema.functions.DecimalValue | Float | Geben Sie einen Dezimalwert ein. | X | |
| Remote access | de.gira.schema.functions.RA.RemoteAccess | RA.RemoteAccess | Gira One – Fern-zugriffsfunktion. | | X |

Auf dem X1 stehen weiterhin die klassischen Funktionstypen `Scene`, `Camera` und `Link` zur Verfügung. Auf dem Gira One werden die entsprechenden Funktionen durch **Function Scene**, **IP-Kamera** (`Camera2`) / **ONVIF-Kamera** und **Link** (`LinkOverlay`) bereitgestellt.

### 10.2 Kanaldefinitionen

**M/O:** Gibt an, ob der Datenpunkt obligatorisch (immer vorhanden) oder optional ist. **R/W/E:** Gibt an, ob der Datenpunkt Lese-, Schreib- oder Ereignisfunktionen unterstützt (Buchstabe wird angezeigt, wenn unterstützt, andernfalls `-`). Wenn ein Datenpunkt die Ereignisübermittlung unterstützt, werden seine Änderungen an registrierte Wert-Callbacks weitergeleitet (siehe Abschnitt 4.7). **Typ** ist der Werttyp des Datenpunkts aus der Definition. Die Spalten **X1** / **Gira One** kennzeichnen, auf welchem Gerät der Kanal verfügbar ist.

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| AudioWithCover | Play | Bool | M | RWE | X | X |
| | Volume | Percent | M | RWE | | |
| | Mute | Bool | O | RWE | | |
| | Previous | Trigger | O | -WE | | |
| | Next | Trigger | O | -WE | | |
| | Title | String | O | R-E | | |
| | Album | String | O | R-E | | |
| | Artist | String | O | R-E | | |
| | Playlist | Byte | O | RWE | | |
| | PreviousPlaylist | Trigger | O | -WE | | |
| | NextPlaylist | Trigger | O | -WE | | |
| | PlaylistName | String | O | R-E | | |
| | Shuffle | Bool | O | RWE | | |
| | Repeat | Bool | O | RWE | | |
| | Cover | Uri | O | R-E | | |
| AudioWithPlaylist | Play | Bool | M | RWE | X | X |
| | Volume | Percent | M | RWE | | |
| | Mute | Bool | O | RWE | | |
| | Previous | Trigger | O | -WE | | |
| | Next | Trigger | O | -WE | | |
| | Title | String | O | R-E | | |
| | Album | String | O | R-E | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Artist | String | O | R-E | | |
| | Playlist | Byte | O | RWE | | |
| | PreviousPlaylist | Trigger | O | -WE | | |
| | NextPlaylist | Trigger | O | -WE | | |
| | PlaylistName | String | O | R-E | | |
| | Shuffle | Bool | O | RWE | | |
| | Repeat | Bool | O | RWE | | |
| Binary | Binary | Binary | M | RWE | X | X |
| Blind2 | Step-Up-Down | UpDown | M | -W- | X | |
| | Up-Down | UpDown | M | -W- | | |
| | Movement | Binary | O | R-E | | |
| | Position | Percent | O | RWE | | |
| | Slat-Position | Percent | O | RWE | | |
| | Upper-Endstop | Bool | O | R-E | | |
| | Lower-Endstop | Bool | O | R-E | | |
| | Status | DWord | O | R-E | | |
| | Lock | Switch | O | RWE | | |
| BlindWithPos | Step-Up-Down | UpDown | M | -W- | X | X |
| | Up-Down | UpDown | M | -W- | | |
| | Movement | Binary | O | R-E | | |
| | Position | Percent | O | RWE | | |
| | Slat-Position | Percent | O | RWE | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| Byte | Byte | Byte | M | RWE | X | |
| Camera | Camera | Binary | O | RWE | X | X |
| DWord | DWord | DWord | M | RWE | X | |
| DimmerRGBW | OnOff | Switch | M | RWE | X | |
| | Brightness | Percent | O | RWE | | |
| | Red | Percent | M | RWE | | |
| | Green | Percent | M | RWE | | |
| | Blue | Percent | M | RWE | | |
| | White | Percent | O | RWE | | |
| DimmerRGBW2 | OnOff | Switch | M | RWE | X | X |
| | Brightness | Percent | O | RWE | | |
| | Red | Percent | M | RWE | | |
| | Green | Percent | M | RWE | | |
| | Blue | Percent | M | RWE | | |
| | White | Percent | O | RWE | | |
| | RGB | DWord | O | RWE | | |
| | RGBW | QWord | O | RWE | | |
| DimmerWhite | OnOff | Switch | M | RWE | X | X |
| | Brightness | Percent | O | RWE | | |
| | Color-Temperature | Temperature | M | RWE | | |
| Float | Float | Float | M | RWE | X | X |
| FunctionScene | Execute | Integer | M | -WE | X | X |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Teach | Integer | O | -WE | | |
| GiraOne.WeatherStationVisu | Wind-Speed | Float | O | R-E | | X |
| | Brightness | Lux | O | R-E | | |
| | Day-Night | Bool | O | R-E | | |
| | Automatic-Day-Night-Operation | Bool | O | R-E | | |
| | Outdoor-Temperature | Temperature | O | R-E | | |
| | Rainfall | Bool | O | R-E | | |
| | Sun-Protection-1 | Bool | O | R-E | | |
| | Sun-Protection-2 | Bool | O | R-E | | |
| | Sun-Protection-3 | Bool | O | R-E | | |
| | Sun-Protection-4 | Bool | O | R-E | | |
| | Wind-Alarm-1 | Bool | O | R-E | | |
| | Wind-Alarm-2 | Bool | O | R-E | | |
| | Rain-Alarm | Bool | O | R-E | | |
| | Frost-Alarm | Bool | O | R-E | | |
| | Fault | Bool | O | R-E | | |
| Hue.Light | OnOff | Switch | M | RWE | X | X |
| | Shift | Percentage | O | -W- | | |
| | Brightness | Percent | O | RWE | | |
| | ColorTemperature | Temperature | O | RWE | | |
| | ShiftColorTemperature | Percentage | O | -W- | | |
| | RelativeBrightnessRelativeColorTemperature | DWord | O | -W- | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | xyY | QWord | O | RWE | | |
| | RGB | DWord | O | RWE | | |
| Integer | Integer | Integer | M | RWE | X | |
| KNX.Dimmer | OnOff | Switch | M | RWE | X | X |
| | Shift | Percentage | O | -W- | | |
| | Brightness | Percent | O | RWE | | |
| KNX.FanCoil | Temperature | Temperature | M | RWE | X | |
| | Set-Point | Temperature | M | RWE | | |
| | OnOff | Switch | M | RWE | | |
| | Mode | Byte | M | RWE | | |
| | Fan-Speed | Byte | O | RWE | | |
| | Vanes-UpDown-Level | Byte | O | RWE | | |
| | Vanes-UpDown-StopMove | Switch | O | RWE | | |
| | Vanes-LeftRight-Level | Byte | O | RWE | | |
| | Vanes-LeftRight-StopMove | Switch | O | RWE | | |
| | Error | Bool | O | R-E | | |
| | Error-Text | Iso88591 | O | R-E | | |
| KNX.HeatingCoolingRelative | Set-Point-Shift | Temperature | M | RWE | X | |
| | Set-Point-Status | Temperature | M | RWE | | |
| | Current | Temperature | M | R-E | | |
| | Mode | Byte | O | -WE | | |
| | Status | Byte | O | R-E | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Presence | Binary | O | RWE | | |
| | Heating | Binary | O | R-E | | |
| | Cooling | Binary | O | R-E | | |
| | Heat-Cool | Bool | O | RWE | | |
| | OnOff | Bool | O | RWE | | |
| KNX.HeatingCoolingSwitchable | Temperature | Temperature | M | RWE | X | X |
| | Set-Point | Temperature | M | RWE | | |
| | Mode | Byte | O | -WE | | |
| | Status | Byte | O | R-E | | |
| | Presence | Binary | O | RWE | | |
| | Heating | Binary | O | R-E | | |
| | Cooling | Binary | O | R-E | | |
| | Heat-Cool | Bool | O | RWE | | |
| | OnOff | Bool | O | RWE | | |
| KNX.WeatherInformation | Wind-Speed | Float | O | R-E | X | |
| | Wind-Direction | Float | O | R-E | | |
| | Brightness | Lux | O | R-E | | |
| | Day-Night | Bool | O | R-E | | |
| | Automatic-Day-Night-Operation | Bool | O | R-E | | |
| | Outdoor-Temperature | Temperature | O | R-E | | |
| | Indoor-Temperature | Temperature | O | R-E | | |
| | Outdoor-Humidity | Percent | O | R-E | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Indoor-Humidity | Percent | O | R-E | | |
| | Air-Pressure | Float | O | R-E | | |
| | Rainfall | Bool | O | R-E | | |
| | Sun-Protection | Bool | O | R-E | | |
| | Wind-Alarm | Bool | O | R-E | | |
| | Rain-Alarm | Bool | O | R-E | | |
| | Frost-Alarm | Bool | O | R-E | | |
| | Fault | Bool | O | R-E | | |
| Link | Link | Binary | O | RWE | X | X |
| Onvif.Camera | Json | Json | O | -WE | X | X |
| | Execute | Json | O | -WE | | |
| | Status | Json | O | R-E | | |
| | CameraInfo | Json | O | R-E | | |
| Percent | Percent | Percent | M | RWE | X | |
| RA.RemoteAccess | Enabled | Bool | M | RWE | | X |
| | User-Access | Bool | M | RWE | | |
| | Installer-Access | Bool | M | RWE | | |
| | Connection-State | Bool | O | R-E | | |
| | Access-State | Bool | O | R-E | | |
| | User-Access-State | Bool | O | R-E | | |
| | Installer-Access-State | Bool | O | R-E | | |
| | Error | Bool | O | R-E | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Connection-Info | String | O | R-E | | |
| | Error-Info | String | O | R-E | | |
| RestClient.BlindWithPos | Step-Up-Down | UpDown | M | -W- | | X |
| | Up-Down | UpDown | M | -W- | | |
| | Movement | Binary | O | R-E | | |
| | Position | Percent | O | RWE | | |
| | Slat-Position | Percent | O | RWE | | |
| RestClient.KNX.Dimmer | OnOff | Switch | M | RWE | | X |
| | Shift | Percentage | O | -W- | | |
| | Brightness | Percent | O | RWE | | |
| RestClient.Switch | OnOff | Switch | M | RWE | | X |
| RoomTemperatureSwitchable | Temperature | Temperature | M | RWE | X | |
| | Set-Point | Temperature | M | RWE | | |
| | OnOff | Switch | O | RWE | | |
| SceneControl | Scene | Scene | M | -W- | X | |
| SceneSet | Execute | Integer | M | -WE | X | |
| | Teach | Integer | O | -WE | | |
| Sonos.Audio | Play | Bool | M | RWE | X | X |
| | Volume | Percent | M | RWE | | |
| | Mute | Bool | O | RWE | | |
| | Previous | Trigger | O | -WE | | |
| | Next | Trigger | O | -WE | | |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Title | String | O | R-E | | |
| | Album | String | O | R-E | | |
| | Artist | String | O | R-E | | |
| | Playlist | Byte | O | RWE | | |
| | PreviousPlaylist | Trigger | O | -WE | | |
| | NextPlaylist | Trigger | O | -WE | | |
| | PlaylistName | String | O | R-E | | |
| | Shuffle | Bool | O | RWE | | |
| | Repeat | Bool | O | RWE | | |
| | Shift-Volume | Percentage | O | -W- | | |
| | Playlists | Json | O | RWE | | |
| | Cover | Uri | O | R-E | | |
| | ValidPlayModes | String | O | R-E | | |
| | TransportActions | String | O | R-E | | |
| | ZoneName | String | O | R-E | | |
| String | String | String | M | RWE | X | |
| Switch | OnOff | Switch | M | RWE | X | X |
| Switch2 | OnOff | Switch | M | RWE | X | |
| | Status | DWord | O | R-E | | |
| | Lock | Switch | O | RWE | | |
| Temperature | Temperature | Temperature | M | RWE | X | |
| ThresholdSetting | Automatic | Bool | M | RWE | X | X |

| Kanal-Typ-URN | Name des Datenpunkts | Typ | M/O | R/W/E | X1 | Gira One |
|---|---|---|---|---|---|---|
| | Threshold | Float | M | RWE | | |
| | Sensor-Value | Float | O | R-E | | |
| Trigger | Trigger | Trigger | M | -WE | X | X |
| Trigger2 | Trigger | Trigger | M | -WE | X | |
| | Status | DWord | O | R-E | | |
| | Lock | Switch | O | RWE | | |
| iot.Trigger | Trigger | Trigger | M | -WE | X | |
| multi.Status | Visu-Data | Json | M | RWE | X | |