www.inside-information.de

Webware RRM Service API

Documentation

Version 1.0

Webware internet solutions GmbH

Teichstraße 14-16
34130 Kas-
sel Ger-
many

Table of Contents

Inside Information Platform API Documentation

1. Overview

The **Inside Information Platform API** extends the ACER REMIT Inside Information Platform with a **technical interface** for automated publication of inside information (UMMs) for:

- Electricity
- Gas
- Other inside information

Main capabilities:

- **Create and publish UMMs** directly via API using a **JSON** contract.
- **Submit corrections** of already published UMMs; only the **latest published message in a thread** can be corrected.
- **Dismiss** (withdraw) a published event via a dedicated endpoint.
- **Read and monitor** published UMMs and office catalog data.
- The **existing web interface** remains unchanged — the API is an additional channel.

Catalog data (market participants & affected assets) must be set up on the platform **before** API publication. On **create**, the API validates market participants and (for Electricity/Gas) the affected asset against your office catalog. Use the **`market_participants`** array when more than one participant is responsible (MoP schema allows multiple `<marketParticipant>` elements). On **correction** (Electricity/Gas), affected asset fields are **not** accepted in the request body — see section 6.1. Unknown catalog values on create return an error; **creating catalog entries (participants and affected assets) is handled by our support team**, not via the API or self-service (see sections 4 and 6.1). Office **admins** may update **Balancing Zone** and **Bidding Zone** defaults for existing affected assets in the web UI only.

Before your first production publish, follow the step-by-step checklist in *section 1.1*.

Important design decisions:

- Via API **only "Save & Publish"** is allowed. Draft-only messages are **not** supported.
- The API accepts **JSON** on input. Field names map **1:1 to the platform's data model**.
- **ACER REMIT XML is generated by the platform on output** (Atom feed and XML download). Clients do **not** send XML.
- Server-managed fields (`message_id`, `publication_datetime`, status, office, threading links, `event_status` on create/correct) are **set by the platform**, never by the client.
- Do **not** send `event_status` in create or correct requests — publications and corrections are always **Active**. The platform may later set the status to **Inactive** automatically once the event has ended (*section 5.2*). Use the **dismiss** endpoint to withdraw an event.
- On **correction** (Electricity/Gas), **`affected_asset_name` and `affected_asset_code` must not be sent** — the affected asset is **fixed for the UMM thread** and is copied from the message you correct. Sending these fields returns `400 VALIDATION_ERROR`. See *section 6.1* and *section 6.11*.
- The API is an **opt-in feature per office**. While disabled, users cannot manage API tokens and API requests are rejected with `403 FORBIDDEN`.
- Only the **latest published** message in a thread can be corrected or dismissed.
- Before any **publish** (create, correct, or dismiss), the platform builds ACER REMIT UMM XML and validates it against the official **XSD** (Electricity/Gas MoP V3, Other V2). Invalid XML is rejected with **`422 XSD_VALIDATION_ERROR`** — nothing is saved. See

section 4 and *section 8*.

Interactive documentation: logged-in users whose office has API enabled can browse the guide and Swagger at **`/api/v1/docs/guide`** and **`/api/v1/docs/swagger`** (menu: **API Docs**). The machine-readable OpenAPI specification is available at **`/api/v1/docs/openapi.yaml`**.

1.1 First-time integration checklist

Use this order when connecting a new client system for the first time.

Step	Action	How to verify
1	Ask support to **enable the API** and the UMM commodities you need (Electricity / Gas / Other) for your office	`GET /api/v1/ping` returns `200` with your token
2	Create an **API token** in the web UI (**API Tokens**)	Token appears in the list; store it in your secret manager
3	Register **market participant(s)** with support (*section 11.1.1*)	`GET /api/v1/catalog/market-participants` lists exact `market_participant_name` / `market_participant_code` pairs
4	For Electricity/Gas: register **affected asset(s)** with support (*section 11.1.2*)	`GET /api/v1/catalog/affected-assets?commodity=electricity` (or `gas`)
5	Integrate on **Test** first (`https://test.inside-information.de/api/v1/`)	Use catalog values **exactly** as returned (case and spelling)
6	Send a **create** request with full JSON body (*section 6.11*)	`201 Created` with `message_id`; or `404` / `VALIDATION_ERROR` / `XSD_VALIDATION_ERROR` — fix payload
7	**Read back** the UMM (`GET …/{message_id}`) and optionally download XML	Response includes `market_participants[]` and `xml_download_url`
8	Switch to **Production** when satisfied	New token recommended; repeat catalog check if office data differs

Market participants in requests

- `market_participants` — array of `{ "name", "code" }` objects (one or more; same names/codes as the catalog). Each entry is resolved against your office catalog.

Corrections: send the **full new state** including the complete `market_participants` list you want on the corrected version. Do **not** send affected asset fields for Electricity/Gas (*section 6.1*).

2. Authentication

Authentication method: Bearer token

Feature activation (per office):

API access must be enabled for an office. Until it is enabled:

- the **API Tokens** screen is not available to that office's users, and
- any API request authenticated with a token of that office is rejected with `403 FORBIDDEN`.

To request activation, contact support (see section 11).

UMM commodities (per office):

Electricity, Gas and Other UMM can be enabled or disabled independently for each office (web UI and API). When a commodity is disabled for your office:

- the corresponding web screens and menu entries are hidden, and
- API requests for that commodity (create, correct, dismiss, list, view, download) are rejected with `403 FORBIDDEN`.

To change commodity availability, contact support (see section 11).

Where to get the token:

Once the feature is enabled, in the web interface under **API Tokens**:

- Create named API tokens (e.g. "SCADA Integration", "Scheduling System").
- See when each token was created and last used.
- Revoke tokens at any time (immediately invalidates API access for that token).

A token is bound to a **user** and therefore to that user's **office**.

How to use the token:

```
Authorization: Bearer <your_api_token>
```

Security recommendations:

- Treat tokens like passwords (store in a secure secret store).
- Use separate tokens per system / environment.
- Rotate tokens regularly or when an integration changes.
- Revoke tokens immediately if you suspect a leak.

3. Base URL, Versioning and Formats

Base URLs:

- **Production**: `https://platform.inside-information.de/api/v1/`
- **Test**: `https://test.inside-information.de/api/v1/`

Versioning:

- All endpoints are prefixed with the version, e.g. `/api/v1/...`.
- Future breaking changes will be introduced under a new version (e.g. `/api/v2/...`).

Request / Response formats:

- **Request body** (create / correct / dismiss): `application/json`.
- **Response**: `application/json`, wrapped in a standard envelope:

Success:

```
{
  "data": { },
  "meta": { }
}
```

Error:

```
{
  "error": {
    "code": "ERROR_CODE",
    "message": "Human-readable message.",
    "details": { }
  }
}
```

- **Character set**: UTF-8
- **Field naming**: snake_case everywhere (request, response, errors).
- **Date / time**: ISO 8601 in **UTC** with a trailing Z, e.g. 2026-06-10T06:00:00Z.

HTTP methods:
All write operations use **`POST`** — create, correct, and dismiss. This follows from the platform's versioning model:

- **Create** publishes a new UMM; the server assigns the message_id (first version _001 in a new thread).
- **Correct** does **not** modify the message at the given URL. It publishes a **new version** in the same thread (_002, _003, …) while the previous version remains in history. Because prior versions are immutable and each correction is a new publication, PUT / PATCH would be semantically wrong.
- **Dismiss** is an **action** that publishes a new version with event_status = Dismissed. The UMM is **not** deleted — it stays in the Atom feed and history — so DELETE does not apply.

Read operations use **`GET`**: list and retrieve UMMs, download XML, and read catalog data.
Status codes on write: create, correct, and dismiss all return **`201 Created`** — each call publishes a new resource/version. There is no separate 200 success path for dismiss.

4. Error Handling

HTTP status codes:

- 201 – Resource created (e.g. new UMM published, corrected, or dismissed — each creates a new thread version).
- 400 – Validation or input error.
- 401 – Authentication failed (missing/invalid token).
- 403 – Not allowed for this token / office.
- 404 – Resource not found. **Always inspect `error.code`** — a 404 may be a generic NOT_FOUND, or a more specific MARKET_PARTICIPANT_NOT_FOUND / AFFECTED_ASSET_NOT_FOUND. Route on error.code, not on the HTTP status alone.
- 409 – Conflict (e.g. correction on a non-latest message, or correction of a dismissed UMM). Inspect error.code to distinguish CONFLICT_NOT_LATEST_IN_THREAD from CONFLICT_ALREADY_DISMISSED.
- 422 – Generated UMM XML failed **ACER XSD** validation (publish blocked). See XSD_VALIDATION_ERROR below.
- 500 – Internal server error.

Error codes (values of error.code):

- AUTH_FAILED – authentication failed.
- FORBIDDEN – token valid but not allowed (API disabled, UMM commodity disabled for office, wrong office, etc.).
- VALIDATION_ERROR – request body invalid. Field-level messages in details.
- XSD_VALIDATION_ERROR – payload passed JSON validation but the UMM XML that would be published does not conform to the ACER REMIT XSD. Nothing is saved. Details

include xsd_errors (list of schema messages).
- NOT_FOUND – requested resource does not exist.
- MARKET_PARTICIPANT_NOT_FOUND – market participant not registered for the office.
- AFFECTED_ASSET_NOT_FOUND – affected asset not registered for the office.
- CONFLICT_NOT_LATEST_IN_THREAD – correction/dismiss attempted on a non-latest message.
- CONFLICT_ALREADY_DISMISSED – correction attempted on a dismissed UMM.
- INTERNAL_ERROR – unexpected server-side error.

Validation errors — details is keyed by field name. Enum fields include allowed values:

```
{
 "error": {
  "code": "VALIDATION_ERROR",
  "message": "Invalid input data.",
  "details": {
   "unavailability_type": {
    "messages": ["Unavailability Type is invalid."],
    "expected": ["Planned", "Unplanned"]
   }
  }
 }
}
```

Other fields use the same shape with messages only (no expected).
XSD validation error (publish blocked):

```
{
 "error": {
  "code": "XSD_VALIDATION_ERROR",
  "message": "UMM XML does not conform to the ACER schema.",
  "details": {
   "xsd_errors": [
    "Element
'{http://www.acer.europa.eu/REMIT/REMITUMMElectricitySchema_V3.xsd}fuelType': This element
is not expected."
   ]
  }
 }
}
```

Fix the request body (or catalog-derived values) and retry. Common causes: missing conditional fields (e.g. fuel_type when event_type is Production unavailability), invalid EIC format, or capacity interval times outside event start/stop.

Unknown market participant / affected asset:

```
{
 "error": {
  "code": "MARKET_PARTICIPANT_NOT_FOUND",
  "message": "Market participant is not registered for your office. Please contact support to have it
added.",
  "details": {
   "market_participant_name": "ACME Trading GmbH",
   "market_participant_code": "B0001064H.DE",
   "contact": {
```

```
      "email": "support@inside-information.de",
      "phone": "+49 (0)561 56014567"
    }
  }
 }
}
```

5. Message and Status Model

5.1 Message ID and Threading

Each UMM is identified by a **`message_id`** (e.g. `00000000000000000000000006_001`):

- **Thread base**: part **before** the underscore.
- **Sequence**: part **after** the underscore (`_001`, `_002`, …).

Rules:

- The **first** published message uses sequence `_001`.
- Every **correction** or **dismiss** creates a **new message** in the same thread with an incremented sequence.
- The previous message is marked as **old** internally.
- Only the **latest published message in a thread** can be corrected or dismissed.

The `message_id` is **server-generated**. Clients never send it in the body; for corrections and dismissals the thread is identified by the `{message_id}` path parameter.

5.2 Event status

`event_status` is **set by the server**, never sent by the client on create or correct. Possible values:

Event status	Meaning	How it is reached
`Active`	The event is published and currently in effect (ongoing or in the future).	Any successful create or correct.
`Inactive`	The event has already ended (its `event_stop` is in the past).	Set **automatically by the platform**, not by the client — see note below.
`Dismissed`	The event has been withdrawn.	A successful dismiss on the latest message in the thread.

- Create and correct always publish with **`event_status` = Active** (set by the server). The client never sends `event_status`.
- **`Inactive` is platform-managed.** A background process runs periodically (roughly hourly) and, once an event's `event_stop` has passed, publishes a **new version** of the thread with `event_status` = `Inactive`. This means an `Active` UMM you published earlier may appear as `Inactive` (with an incremented sequence) on later reads — without any action on your side. Do not treat `Active` as permanent until dismissal.
- Because this transition is a new thread version, the previously published `Active` version remains unchanged and accessible (immutability is preserved).
- To **dismiss** an event, call `POST /umm/{commodity}/{message_id}/dismiss` on the **latest** message in the thread.
- A **dismissed** UMM **cannot be corrected** (409 `CONFLICT_ALREADY_DISMISSED`).

6. UMM Endpoints

All UMM endpoints are under /api/v1/umm/ and require authentication.

6.1 Common Request Concepts

Identification by business keys (no internal IDs):

Request field	Purpose	Resolution
market_participants[]	One or more { name, code } objects	Each pair resolved in the **office catalog** (must be pre-registered).
affected_asset_name + affected_asset_code	Affected asset (Electricity/Gas **create** only)	Resolved in the **office catalog**.

Pre-registration required (via support). The API does **not** create catalog entries. Before your first UMM request, the following must already exist in the platform **for your office**:

- **At least one market participant** — via market_participants (≥1 entry) — required for **all** commodities.
- **`affected_asset_name` + `affected_asset_code`** — required on **create** for **Electricity** and **Gas** (not used for Other). **Not accepted on correction** — the asset is inherited from the thread (see below).

If you send a name/code pair that is not registered for your office, the request is rejected:

- 404 MARKET_PARTICIPANT_NOT_FOUND — unknown market participant.
- 404 AFFECTED_ASSET_NOT_FOUND — unknown affected asset (Electricity/Gas).

Adding entries is done by our support team (not self-service, not via API). Contact support with the participant and/or asset details; see *section 11.1* for example emails. After we add them, use the catalog endpoints (section 7.4–7.5) or the web UI to confirm the exact name / code values before integrating.

Affected asset zone defaults (web UI only). Creating or deleting affected assets is not available to office users. Users with the **admin** role for an office may update **Balancing Zone** (Gas) and **Bidding Zone** (Electricity) defaults on existing catalog entries via **Affected assets** in the web interface. The API has no endpoint to create or modify catalog entries; UMM **create** requests must reference affected_asset_name / affected_asset_code exactly as registered.

Create vs correction (Electricity / Gas).

Topic	**Create (**POST /umm/{commodity}**)**	**Correct** (POST /umm/{commodity}/{message_id}/correct)
Request schema (OpenAPI)	{Commodity}UmmWrite	{Commodity}UmmCorrect
market_participants	**Required** (≥1 entry)	**Required** — full list for the new version (not a diff). May differ from the previous version (see note below).
affected_asset_name / affected_asset_code	**Required** — resolved against catalog	**Must not be sent** — copied from the corrected message / thread
All other writable fields	Full new state	Full new state (same as create, except affected asset)

Topic	Create (POST /umm/{commodity})	Correct (POST /umm/{commodity}/{message_id}/correct)
Web UI	Asset selectable	Asset **disabled** when correcting a published UMM

If you need to disclose inside information for a **different** affected asset, publish a **new UMM** (new thread), not a correction.

Market participants on correction. Unlike the affected asset, the `market_participants` list is **accepted on correct** and may differ from the previous version — each entry is re-resolved against your office catalog. This supports legitimate cases such as fixing an incorrect participant or adjusting the set of responsible parties for a jointly-operated asset. However, the participant list identifies **who is responsible for disclosing the event**; replacing it with an entirely unrelated participant usually signals a **different event**, which should be a **new UMM (new thread)** rather than a correction. The platform does not block this, so it is the client's responsibility to keep corrections tied to the same underlying event.

Server-managed fields (never sent by the client):

`message_id`, `publication_datetime`, status, `office_id`, `event_status` (on create/correct), threading links, audit timestamps.

On **correction**, also **never send**: `affected_asset_name`, `affected_asset_code` (Electricity/Gas).

Common body fields (Electricity and Gas — create):

Field	Type	Required on create	Notes
`event_type`	string	yes	Commodity-specific enum.
`unavailability_type`	string	yes	`Planned` or `Unplanned`.
`event_start`	string	yes	ISO 8601 UTC.
`event_stop`	string	yes	ISO 8601 UTC, must be ≥ `event_start`.
`unit_measure`	string	yes	Commodity-specific enum.
`unavailability_reason`	string	yes	Free text.
`remarks`	string	no	Free text.
`market_participants`	object[]	yes	At least one `{ "name", "code" }` (catalog). See *section 6.11*.
`affected_asset_name`	string (≤50)	yes	See identification table. **Omit on correct.**
`affected_asset_code`	string (16)	yes	EIC code. **Omit on correct.**

Machine-readable schemas: field definitions are in `*docs/openapi.yaml*`:

- **Create**: ElectricityUmmWrite, GasUmmWrite, OtherUmmWrite
- **Correct**: ElectricityUmmCorrect, GasUmmCorrect, OtherUmmCorrect (Electricity/Gas correct schemas **exclude** affected asset)
- **Dismiss**: UmmDismissRequest

Human-readable reference: *section 6.11*.

6.11 JSON request body schemas (reference)

OpenAPI defines **separate** schemas for **create** and **correct**:

Operation	Electricity	Gas	Other
Create	ElectricityUmmWrite	GasUmmWrite	OtherUmmWrite
Correct	ElectricityUmmCorrect	GasUmmCorrect	OtherUmmCorrect

For **create** and **correct**, the body is the **full new state** of the message you want published, not a field-level diff.

> ⚠ **Correction is a full replacement, not a partial update.** A correct request must include **every field you want the new version to have** (except fields listed below as thread-fixed or server-managed). Any writable field you omit is **not** carried over from the previous version. Do **not** send only the changed fields.

Exceptions on correct (Electricity / Gas):

- **`affected_asset_name` / `affected_asset_code`** — do **not** send. The platform keeps the affected asset from the UMM thread. If present in the JSON body → 400 VALIDATION_ERROR.
- **`event_status`** — do **not** send (always Active on correct; the platform may later set Inactive automatically — see *section 5.2*; use dismiss to withdraw).

`market_participants` on correct — **is** sent (full list, like create) and **may differ** from the previous version; each entry is re-resolved against the catalog. Keep it tied to the same event — a wholesale change of participant usually means a new thread (see *section 6.1*).

Do not send on create/correct: message_id, publication_datetime, event_status, office_id, threading/audit fields.

Do not send on correct only (Electricity/Gas): affected_asset_name, affected_asset_code.

Shared types

Type	Format / values
ISO datetime	2026-06-10T06:00:00Z (UTC, trailing Z)
unavailability_type	Planned, Unplanned
EIC zone code	string, exactly 16 characters. Send the code **verbatim**, including any filler dashes (e.g. 10YDE-VE-------2). Do not strip or trim the dashes.
EIC asset code	string, exactly 16 characters (affected_asset_code)

Market participant (all commodities):

Field	Type	Required	Constraints
market_participants	array	**yes**	At least one object with name + code. Every pair must match a participant **already registered for your office**. Order is preserved in published XML.

Example (two participants):

```
"market_participants": [
  { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" },
  { "name": "ACME Trading GmbH", "code": "11X0000000012345" }
]
```

Read responses (GET / list) include market_participants — an array of { "market_participant_name", "market_participant_code" } (full list, same order as in the published XML).

Affected asset (Electricity and Gas — **create only**):

Field	Type	Required	Constraints
affected_asset_name	string	yes on create	Must match an asset **already registered for your office** (added by support); max 50 characters
affected_asset_code	string	yes on create	16-character EIC W/T/Z/A/V code from the catalog entry

On **correction**, omit both fields; the response and published XML still contain the asset from the thread (same as GET on any message in that thread).

Zone defaults (balancing_zones, bidding_zones) on the catalog entry may be maintained by office admins in the web UI; they are **not** sent in UMM API requests (zones are part of the UMM body per section 6.2 / 6.3).

If either pair is missing from your office catalog → 404 with support contact in details (see section 4). Request addition via support (section 11); the API cannot register participants or assets.

Electricity — ElectricityUmmWrite (create)

Field	Type	Required	Allowed values / notes
market_participants	array	yes	See *Market participant*
affected_asset_name	string	yes	max 50
affected_asset_code	string	yes	16-char EIC

Field	Type	Required	Allowed values / notes
event_type	string	yes	Production unavailability, Transmission unavailability, Consumption unavailability, Other unavailability
unavailability_type	string	yes	Planned, Unplanned
event_start	string	yes	ISO datetime UTC
event_stop	string	yes	ISO datetime UTC, ≥ event_start
unit_measure	string	yes	MW
installed_capacity	number	yes	
fuel_type	string	conditional	**Required** when event_type = Production unavailability. One of: Biomass, Fossil Brown coal/Lignite, Fossil Coal-derived gas, Fossil Gas, Fossil Hard coal, Fossil Oil, Fossil Oil shale, Fossil Peat, Geothermal, Hydro Pumped Storage, Hydro Run-of-river and poundage, Hydro Water Reservoir, Marine, Nuclear, Other renewable, Solar, Waste, Wind Offshore, Wind Onshore, Other
bidding_zones	string[]	yes	min 1 item; each element 16-char EIC zone code
capacity_intervals	object[]	yes	min 1 item; see rules below
unavailability_reason	string	yes	Free text
remarks	string	no	max 500 characters

`capacity_intervals[]` rules:

- At least **one** interval is required; a single interval covering the whole event is valid.
- Each interval needs interval_start < interval_stop.
- Intervals **must not overlap** each other.
- Each interval must fall **within** [event_start, event_stop].
- Gaps between intervals are allowed (intervals need not be contiguous).

- A violation returns `400 VALIDATION_ERROR` with the offending interval indicated in `details`.

`capacity_intervals[]` item:

Field	Type	Required
interval_start	string (ISO UTC)	yes
interval_stop	string (ISO UTC)	yes
unavailable_capacity	number	yes
available_capacity	number	yes

Example (create):

```
{
 "market_participants": [
  { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
 ],
 "affected_asset_name": "Block A Power Plant",
 "affected_asset_code": "11WXYZ0000000012",
 "event_type": "Production unavailability",
 "unavailability_type": "Planned",
 "event_start": "2026-06-10T06:00:00Z",
 "event_stop": "2026-06-12T18:00:00Z",
 "unit_measure": "MW",
 "installed_capacity": 800,
 "fuel_type": "Fossil Gas",
 "bidding_zones": ["10YDE-VE-------2", "10YDE-EON------1"],
 "capacity_intervals": [
  {
   "interval_start": "2026-06-10T06:00:00Z",
   "interval_stop": "2026-06-11T06:00:00Z",
   "unavailable_capacity": 300,
   "available_capacity": 500
  }
 ],
 "unavailability_reason": "Technical failure",
 "remarks": "Optional free text"
}
```

Electricity — `ElectricityUmmCorrect` (correct)

Same fields as `ElectricityUmmWrite`, **except** omit `affected_asset_name` and `affected_asset_code`. OpenAPI: `ElectricityUmmCorrect`.

Example (correct — note: no affected asset fields):

```
{
 "market_participants": [
  { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
 ],
 "event_type": "Production unavailability",
 "unavailability_type": "Planned",
 "event_start": "2026-06-10T06:00:00Z",
 "event_stop": "2026-06-13T18:00:00Z",
 "unit_measure": "MW",
 "installed_capacity": 800,
 "fuel_type": "Fossil Gas",
```

```
  "bidding_zones": ["10YDE-VE-------2", "10YDE-EON------1"],
  "capacity_intervals": [
    {
      "interval_start": "2026-06-10T06:00:00Z",
      "interval_stop": "2026-06-11T06:00:00Z",
      "unavailable_capacity": 300,
      "available_capacity": 500
    }
  ],
  "unavailability_reason": "Technical failure — extended outage",
  "remarks": "Corrected end date"
}
```

The published correction keeps `affected_asset_name` / `affected_asset_code` from the thread (visible in GET/list responses).

Gas — `GasUmmWrite` (create)

Field	Type	Required	Allowed values / notes
`market_participants`	array	yes	See *Market participant*
`affected_asset_name`	string	yes	max 50
`affected_asset_code`	string	yes	16-char EIC
`event_type`	string	yes	`Offshore pipeline unavailability`, `Transmission system unavailability`, `Storage unavailability`, `Storage facility unavailability`, `Injection unavailability`, `Withdrawal unavailability`, `Gas treatment plant unavailability`, `Regasification plant unavailability`, `Compressor station unavailability`, `Gas production field unavailability`, `Import contract curtailment`, `Consumption unavailability`, `Other unavailability`
`unavailability_type`	string	yes	`Planned`, `Unplanned`
`event_start`	string	yes	ISO datetime UTC
`event_stop`	string	yes	ISO datetime UTC, ≥ `event_start`
`unit_measure`	string	yes	`kWh/h`, `kWh/d`, `GWh/d`, `GWh`, `TWh`,

Field	Type	Required	Allowed values / notes
			mcm/d
unavailable_capacity	number	yes	In unit_measure
available_capacity	number	yes	In unit_measure
technical_capacity	number	yes	In unit_measure
balancing_zones	string[]	yes	min 1 item; each element 16-char EIC zone code
direction	string	no	Entry, Exit. Relevant for flow/transmission-type events (e.g. entry/exit points); omit when not applicable to the event type.
unavailability_reason	string	yes	Free text
remarks	string	no	max 500 characters

Example (create):

```
{
 "market_participants": [
  { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
 ],
 "affected_asset_name": "Storage Site North",
 "affected_asset_code": "11WXYZ0000000012",
 "event_type": "Transmission system unavailability",
 "unavailability_type": "Unplanned",
 "event_start": "2026-06-10T06:00:00Z",
 "event_stop": "2026-06-12T18:00:00Z",
 "unit_measure": "kWh/d",
 "unavailable_capacity": 1200,
 "available_capacity": 800,
 "technical_capacity": 2000,
 "balancing_zones": ["10Y1001A1001A59J"],
 "direction": "Entry",
 "unavailability_reason": "Maintenance",
 "remarks": "Optional"
}
```

Gas — GasUmmCorrect (correct)

Same as GasUmmWrite, **without** affected_asset_name / affected_asset_code. OpenAPI: GasUmmCorrect.

Other — `OtherUmmWrite` (create / correct)

Other inside information has **no** affected asset, capacity fields, or zones.

> **Note:** unlike Electricity and Gas (where `remarks` is optional), for **Other** UMM `remarks` is **required** — it is the free-text body that describes the inside information.

Field	Type	Required	Allowed values / notes
`market_participants`	array	yes	See *Market participant*
`event_start`	string	yes	ISO datetime UTC
`event_stop`	string	no	ISO datetime UTC when present; must be ≥ `event_start`
`remarks`	string	yes	Free-text description of the event; max 1000 characters

Example (create or correct — Other has no affected asset):

```
{
  "market_participants": [
    { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
  ],
  "event_start": "2026-06-10T06:00:00Z",
  "event_stop": "2026-06-12T18:00:00Z",
  "remarks": "Required free-text description of the inside information."
}
```

Dismiss — `UmmDismissRequest` (all commodities)

Used with `POST /api/v1/umm/{electricity|gas|other}/{message_id}/dismiss`.

Field	Type	Required	Notes
`remarks`	string	no	Optional reason for withdrawal. Max **500** characters (Electricity, Gas). Max **1000** characters (**Other** — if sent, replaces the event body `remarks` on the dismissed version; if omitted, the previous text is kept).

Example:

```
{
  "remarks": "Event cancelled"
}
```

Empty body `{}` is also accepted.

6.2 Create Electricity UMM

Method: POST **URL**: /api/v1/umm/electricity
Electricity-specific fields (full create schema: ElectricityUmmWrite in *section 6.11*; correction: *6.3*):

Field	Type	Required	Notes
event_type	string	yes	Production unavailability, Transmission unavailability, Consumption unavailability, Other unavailability.
unit_measure	string	yes	MW.
installed_capacity	number	yes	Installed capacity.
fuel_type	string	conditional	Required when event_type = Production unavailability.
bidding_zones	string[]	yes	At least one EIC zone code.
capacity_intervals	object[]	yes	At least one interval; must not overlap.

Request example:

```
{
 "market_participants": [
  { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
 ],
 "affected_asset_name": "Block A Power Plant",
 "affected_asset_code": "11WXYZ0000000012",
 "event_type": "Production unavailability",
 "unavailability_type": "Planned",
 "event_start": "2026-06-10T06:00:00Z",
 "event_stop": "2026-06-12T18:00:00Z",
 "unit_measure": "MW",
 "installed_capacity": 800,
 "fuel_type": "Fossil Gas",
 "bidding_zones": ["10YDE-VE-------2", "10YDE-EON------1"],
 "capacity_intervals": [
  {
   "interval_start": "2026-06-10T06:00:00Z",
   "interval_stop": "2026-06-11T06:00:00Z",
   "unavailable_capacity": 300,
   "available_capacity": 500
  }
 ],
 "unavailability_reason": "Technical failure",
 "remarks": "Optional free text"
}
```

Successful response (201 Created):


```
{
 "data": {
  "message_id": "00000000000000000000000123_001",
  "thread_base": "000000000000000000000000123",
  "status": "PUBLISHED",
  "published_at": "2026-06-09T10:00:00Z",
  "event_status": "Active",
  "event_type": "Production unavailability"
 },
 "meta": {
  "environment": "test",
  "commodity": "electricity"
 }
}
```


Common errors on create: 404 MARKET_PARTICIPANT_NOT_FOUND / AFFECTED_ASSET_NOT_FOUND (catalog mismatch), 400 VALIDATION_ERROR (missing/invalid fields), 422 XSD_VALIDATION_ERROR (XML schema check failed — fix payload and retry). See *section 4*.

Method: POST **URL**: /api/v1/umm/electricity/{message_id}/correct

Request body: ElectricityUmmCorrect — the **new complete state** for the correction, **without** affected_asset_name / affected_asset_code. See *section 6.11*.

Only the **latest published message** in the thread can be corrected.

Possible errors:

- 409 CONFLICT_NOT_LATEST_IN_THREAD — the target message is not the latest in its thread.
- 409 CONFLICT_ALREADY_DISMISSED — the thread has been dismissed and can no longer be corrected.
- 400 VALIDATION_ERROR — body invalid (full new state required; do not send affected asset or event_status).
- 422 XSD_VALIDATION_ERROR — JSON was valid but generated ACER XML failed XSD checks; nothing was saved (see *section 4*).
- 404 NOT_FOUND — the {message_id} does not exist for your office.

6.4 Dismiss Electricity UMM

Method: POST **URL**: /api/v1/umm/electricity/{message_id}/dismiss

Request body (optional):


```
{
 "remarks": "Optional reason for dismissal"
}
```


Creates a new thread version with event_status = Dismissed. Only the latest published message can be dismissed.

Successful response (201 Created):


```
{
 "data": {
  "message_id": "00000000000000000000000123_002",
  "thread_base": "000000000000000000000000123",
  "status": "PUBLISHED",
  "published_at": "2026-06-09T12:00:00Z",
```

```
    "event_status": "Dismissed",
    "previous_message_id": "00000000000000000000000123_001"
  },
  "meta": {
    "environment": "prod",
    "commodity": "electricity"
  }
}
```


6.5 Create Gas UMM

Method: POST **URL**: /api/v1/umm/gas

Gas-specific fields (full schema: *section 6.11*):

Field	Type	Required	Notes
event_type	string	yes	Gas event type enum (see OpenAPI / interactive docs).
unit_measure	string	yes	kWh/h, kWh/d, GWh/d, GWh, TWh, or mcm/d.
unavailable_capacity	number	yes	In unit_measure.
available_capacity	number	yes	In unit_measure.
technical_capacity	number	yes	In unit_measure.
balancing_zones	string[]	yes	At least one EIC zone code.
direction	string	no	Entry or Exit.

6.6 Correct Gas UMM

Method: POST **URL**: /api/v1/umm/gas/{message_id}/correct

Same rules as *Electricity correction (6.3)*. Request body: GasUmmCorrect (no affected asset fields).

6.7 Dismiss Gas UMM

Method: POST **URL**: /api/v1/umm/gas/{message_id}/dismiss

Same rules as Electricity dismiss (section 6.4).

6.8 Create Other UMM

Method: POST **URL**: /api/v1/umm/other

Fields (full schema: *section 6.11*):

Field	Type	Required	Notes
market_participants	array	yes	See *Market participant*.
event_start	string	yes	ISO 8601 UTC.
event_stop	string	no	ISO 8601 UTC when present.
remarks	string	yes	Free-text description;

Field	Type	Required	Notes
			max 1000 characters.

No affected asset, capacity, or zones.

6.9 Correct Other UMM

Method: POST **URL**: /api/v1/umm/other/{message_id}/correct
Request body: OtherUmmCorrect — same fields as create (Other has no affected asset). Full new state, not a diff.

6.10 Dismiss Other UMM

Method: POST **URL**: /api/v1/umm/other/{message_id}/dismiss

7. Read / Monitoring & Catalog Endpoints

7.1 List UMMs

Method: GET **URL**: /api/v1/umm/{commodity} {commodity} ∈ electricity | gas | other
Query parameters (optional): message_id, thread_base, from, to, status, page, per_page.
Each item in data contains the **same fields as GET** /api/v1/umm/{commodity}/{message_id} (full UMM JSON including xml_download_url), plus **`is_latest_in_thread`** to identify the current version in a correction thread.
Pagination: list responses return paging info in meta. Iterate until page × per_page ≥ total (or until a page returns fewer than per_page items):

```
{
 "data": [ /* … UMM items … */ ],
 "meta": {
  "page": 1,
  "per_page": 50,
  "total": 137
 }
}
```

7.2 Get UMM by message_id

Method: GET **URL**: /api/v1/umm/{commodity}/{message_id}
Returns the same full UMM JSON as a list item (without is_latest_in_thread).
Example response (Electricity — abbreviated):

```
{
 "data": {
  "message_id": "00000000000000000000000123_001",
  "thread_base": "00000000000000000000000123",
  "status": "PUBLISHED",
  "published_at": "2026-06-09T10:00:00Z",
  "event_status": "Active",
  "market_participants": [
   {
    "market_participant_name": "Webware Internet Solutions GmbH",
    "market_participant_code": "B0001064H.DE"
   },
   {
    "market_participant_name": "ACME Trading GmbH",
    "market_participant_code": "11X0000000012345"
```

```
    }
   ],
   "affected_asset_name": "Block A Power Plant",
   "affected_asset_code": "11WXYZ0000000012",
   "event_type": "Production unavailability",
   "event_start": "2026-06-10T06:00:00Z",
   "event_stop": "2026-06-12T18:00:00Z",
   "xml_download_url": "/api/v1/umm/electricity/00000000000000000000000123_001/download"
  },
  "meta": {
   "environment": "test",
   "commodity": "electricity"
  }
}
```


7.3 Download UMM XML

Method: GET **URL**: /api/v1/umm/{commodity}/{message_id}/download
Response: application/xml — ACER REMIT UMM XML generated by the platform.

7.4 List Market Participants

Method: GET **URL**: /api/v1/catalog/market-participants
Query: q, page, per_page.

7.5 List Affected Assets

Method: GET **URL**: /api/v1/catalog/affected-assets
Read-only list of affected assets registered for your office (including zone defaults). To **add** assets or change name/EIC, contact support (section 11). Office admins may update **Balancing Zone** / **Bidding Zone** defaults in the web UI under **Affected assets**.
Query: commodity (electricity or gas), q, page, per_page.

7.6 Health check

Method: GET **URL**: /api/v1/ping
Verifies token validity and that API is enabled for the office.

8. Atom Feed and Publication Behaviour

The public Atom feed remains the **primary publication channel**.

- Only **published** UMMs appear in Atom feeds.
- API-created UMMs behave exactly like web-created UMMs.
- Corrections and dismissals create new Atom entries; older entries remain in history.
- Before any publish (API or web UI), the platform generates ACER REMIT UMM XML and validates it against the official **XSD** (Electricity/Gas MoP V3, Other V2). If validation fails, the operation is rejected with **`422 XSD_VALIDATION_ERROR`** and **no database row is written**. Fix the payload and retry. Downloading XML via GET …/download always returns XML that passed this check at publish time.
- Multiple market participants in market_participants[] appear as multiple <marketParticipant> elements in the published XML and Atom entry, in the same order as the JSON array.

9. Rate Limiting and Usage

Note — rate limiting (planned): Per-token rate limiting with `X-RateLimit-Limit`, `X-RateLimit-Remaining`, `X-RateLimit-Reset` headers and HTTP `429` is on the roadmap but **not implemented** in the current API release. Do not rely on these headers today. Contact support (section 11) if you expect high request volume.

- **JSON body size** limited to a reasonable maximum (web server / PHP limits); oversized bodies are rejected with `400`. Contact support if your payloads approach the limit.

Best practices:

- **All write operations are non-idempotent.** Create, correct, and dismiss each publish a **new** version. Re-sending the same request after a successful publish does **not** overwrite the previous one — it creates a **duplicate** (e.g. two UMMs for the same event, or an extra correction in the thread).
- **Retrying safely after a network failure:** if you get no response (timeout, dropped connection), do **not** blindly re-POST. First check whether the previous attempt succeeded (see next two bullets). Re-send only if the UMM or version you expected is **not** there.
- **Retry check — create:** search by time window — `GET /api/v1/umm/{commodity}?from=…&to=…` — or by `message_id` if you already received it.
- **Retry check — correct / dismiss:** list the thread — `GET /api/v1/umm/{commodity}?thread_base={thread_base}` — and confirm whether the expected new `message_id` or `event_status` is already present.
- Implement exponential backoff on **`5xx`** responses.
- Prefer **correction** endpoints over publishing a second UMM for the same underlying event.

10. Example Usage

10.1 Create Electricity UMM

```
curl -X POST "https://test.inside-information.de/api/v1/umm/electricity" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "market_participants": [
      { "name": "Webware Internet Solutions GmbH", "code": "B0001064H.DE" }
    ],
    "affected_asset_name": "Block A Power Plant",
    "affected_asset_code": "11WXYZ0000000012",
    "event_type": "Production unavailability",
    "unavailability_type": "Planned",
    "event_start": "2026-06-10T06:00:00Z",
    "event_stop": "2026-06-12T18:00:00Z",
    "unit_measure": "MW",
    "installed_capacity": 800,
    "fuel_type": "Fossil Gas",
    "bidding_zones": ["10YDE-VE-------2"],
    "capacity_intervals": [
      {
        "interval_start": "2026-06-10T06:00:00Z",
        "interval_stop": "2026-06-11T06:00:00Z",
        "unavailable_capacity": 300,
        "available_capacity": 500
      }
```

```
  ],
  "unavailability_reason": "Technical failure"
}'
```

10.2 Dismiss a Gas UMM

```
curl -X POST \
  "https://test.inside-information.de/api/v1/umm/gas/00000000000000000000000055_001/dismiss"
\
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{ "remarks": "Event cancelled" }'
```

11. Support

For API feature availability, onboarding, **registering market participants and affected assets**, and technical support:

- Website: https://inside-information.de/
- Email: support@inside-information.de
- Phone: as published in the platform contact section (+49 (0)561 56014567 at time of writing).

The API endpoints become available to an office only after the API feature is activated for that account.

11.1 Catalog registration by email

The API and self-service **cannot** create market participants or affected assets. In the web UI, open **Participants** or **Affected assets**, choose the registration template link, complete the prefilled form, and click **Send request** — the platform emails **`support@inside-information.de`** on your behalf (via the job queue). You can also copy the example templates below if you prefer to send email manually.

After support confirms registration, verify the exact values via:

- Web UI: **Participants** / **Affected assets**, or
- API: GET /api/v1/catalog/market-participants and GET /api/v1/catalog/affected-assets

Use those exact name / code strings from the catalog in UMM **create** and **correct** requests inside the market_participants array. For Electricity/Gas, also use affected_asset_name and affected_asset_code on **create** only. Corrections do not accept affected asset fields — the asset stays tied to the UMM thread.

You can register **multiple** market participants for one office; list them all in market_participants when a UMM involves more than one responsible party.

Office admins may later update **Balancing Zone** / **Bidding Zone** defaults for existing affected assets in the web UI (section 6.1); include suggested zone defaults in the registration email when known.

11.1.1 Market participant

Subject: IIP creation request — Market participant — [Office company name]

Email body (example):

```
Dear Inside Information Support,

Please register a new market participant in our office catalog on the Inside Information Platform.

Office (as registered on the platform): [Company name]
Target environment: [Test | Prod | Test and Prod]
Contact person: [First name Last name]
```

```
Contact email: [your@company.com]
Platform username (optional): [username]

Market participant(s) (maps to API field `market_participants` — repeat the block for each
participant):
  - Market participant name: ACME Trading GmbH
  - Market participant type: EIC
  - Market participant ID / code: B0001064H.DE
  - (optional second participant)
  - Market participant name: Partner Energy AG
  - Market participant type: EIC
  - Market participant ID / code: 11X0000000098765

Registered address:
- Postal code: 34117
- Country: DE
- City: Kassel
- Address: Wilhelmshöher Allee 123
- Additional address (optional): Building B

Contact at market participant (ACER REMIT):
- Gender: Mr.
- First name: Max
- Last name: Mustermann
- Organisation unit: Compliance
- Job title: REMIT Officer
- Phone: +49 561 1234567
- Fax (optional): +49 561 1234568
- Email: remit@acme-trading.example

Best regards,
[Name]
[Company]
```

11.1.2 Affected asset — Gas only

Subject: IIP creation request — Affected asset (Gas) — [Office company name]

Email body (example):

```
Dear Inside Information Support,

Please register a new affected asset (Gas) in our office catalog on the Inside Information Platform.

Office (as registered on the platform): [Company name]
Target environment: [Test | Prod | Test and Prod]
Contact person: [First name Last name]
Contact email: [your@company.com]

Affected asset (maps to API fields affected_asset_name / affected_asset_code):
- Affected asset name: Storage Site North
- Affected asset EIC code (16 characters): 11WXYZ0000000012
- Commodity type: Gas

Default values for UMM forms (catalog defaults; may be changed per event in UMM):
- Unit of measurement: GWh/d  (allowed: kWh/h, kWh/d, GWh/d, GWh, TWh, mcm/d)
- Technical capacity: 500
- Balancing zone(s) (16-character EIC code(s), one per line):
  10Y1001A1001A59J

Best regards,
```

[Name]
[Company]

11.1.3 Affected asset — Electricity only

Subject: IIP creation request — Affected asset (Electricity) — [Office company name]

Email body (example):

Dear Inside Information Support,

Please register a new affected asset (Electricity) in our office catalog on the Inside Information Platform.

Office (as registered on the platform): [Company name]
Target environment: [Test | Prod | Test and Prod]
Contact person: [First name Last name]
Contact email: [your@company.com]

Affected asset (maps to API fields affected_asset_name / affected_asset_code):
- Affected asset name: Block A Power Plant
- Affected asset EIC code (16 characters): 11WXYZ0000000020
- Commodity type: Electricity

Default values for UMM forms (catalog defaults; may be changed per event in UMM):
- Unit of measurement: MW (Electricity: MW only)
- Installed capacity: 850
- Bidding zone(s) (16-character EIC code(s), one per line):
  10YDE-VE-------2

Best regards,
[Name]
[Company]

11.1.4 Affected asset — Gas and Electricity

Subject: IIP creation request — Affected asset (Gas and Electricity) — [Office company name]

Email body (example):

Dear Inside Information Support,

Please register a new affected asset (Gas and Electricity) in our office catalog on the Inside Information Platform.

Office (as registered on the platform): [Company name]
Target environment: [Test | Prod | Test and Prod]
Contact person: [First name Last name]
Contact email: [your@company.com]

Affected asset (maps to API fields affected_asset_name / affected_asset_code):
- Affected asset name: Combined Cycle Unit East
- Affected asset EIC code (16 characters): 11WXYZ0000000038
- Commodity type: Gas and Electricity

Gas section — default values for UMM forms:
- Unit of measurement: GWh/d (allowed: kWh/h, kWh/d, GWh/d, GWh, TWh, mcm/d)
- Technical capacity: 1200
- Balancing zone(s) (16-character EIC code(s), one per line):
  10Y1001A1001A59J

Electricity section — default values for UMM forms:

```
- Unit of measurement: MW  (Electricity: MW only)
- Installed capacity: 650
- Bidding zone(s) (16-character EIC code(s), one per line):
  10YDE-VE-------2

Best regards,
[Name]
[Company]
```