DATAHUB 2.0 USER GUIDE

1. INTRODUCTION

Sentinum Datahub 2.0 provides reliable MQTT access to Sentinum sensor telemetry, configuration, and diagnostics. It is the evolution of the legacy broker under data.sentinum.de and is available at broker.sentinum.de.

Migration notice: The legacy broker data.sentinum.de is deprecated and will be shut down no later than the end of 2026. Migration to the new broker broker.sentinum.de is strongly recommended.

1.1 FEATURES

- High reliability and security through a clustered architecture hosted in the European Union.
- Designed to handle large device fleets.
- Support for MQTT 3.0, 3.1, and 5.0.
- Strict ACL rules so sensors and device data are only accessible to authorized owners.
- Shadow-based device parameterization for simple integration with external applications.
- Customer-specific interface and protocol adaptations, including transformations, direct database writes, and bridging.
- Public TLS certificate handling without service interruptions during certificate renewal.
- Client monitoring for improved security and malfunction detection.

1.2 BACKWARD COMPATIBILITY

- A read-only mirror of the legacy topic tree is available under devices/# to support step-by-step migration.
- Existing client credentials were migrated to broker.sentinum.de.
- MQTT 3.0 and 3.1 clients remain supported.

1.3 LIMITATIONS

- The broker requires MQTTS with TLS on port 8883, or secure WebSockets on port 8084.
- Plain MQTT on port 1883 is no longer available.

- ACL rules are stricter than before. Clients can publish only to explicitly permitted topics.
- Shadow management and device info are available for cellular-based sensors with major version 2 or above.

1.4 CREDENTIALS

Each client authenticates with a username and password issued by Sentinum on request. These credentials are independent of my.sentinum.de user accounts.

The same username and password can be used by multiple clients. Each client should still use a unique MQTT Client ID.

2. TOPIC TREE AND PERMISSIONS

Datahub 2.0 uses a structured topic tree that combines telemetry, configuration management, and device status.

2.1 BASE TOPIC

All device-specific topics use this base topic:

datahub/<customerName>/<deviceBaseType>/<deviceSubType>/<deviceId>

Placeholders:

PLACEHOLDER	DESCRIPTION
<CUSTOMERNAME>	Customer name as used in my.sentinum.de, lowercased and with spaces replaced by underscores.
<DEVICEBASETYPE>	Sensor series, for example apollon, febris, juno, or helios.
<DEVICESUBTYPE>	Sub-series name. This is reserved for special use cases, customization, and future updates. It is safe to mirror <deviceBaseType> here.
<DEVICEID>	Unique device ID in the shape <series>_<transport>_<identifier>, for example apollon_mioty_FCA84A01000036C7, apollon_cellular_355025930266075, or febris_lora_FCA84A0900000134.

The following sections use <baseTopic> as shorthand for the full base topic.

2.2 READ-ONLY TOPICS

Clients can subscribe to these topics. Publishing to them is denied and can cause the client to be disconnected.

TOPIC	DESCRIPTION
<BASETOPIC>/UP	Real-time telemetry. The device publishes sensor data periodically or when events occur, depending on its configuration.
<BASETOPIC>/REPORTED_SHADOW	Full configuration currently applied on the device. If a desired shadow delta exists, the new configuration appears here after the device receives it and confirms that it was applied successfully.
<BASETOPIC>/DESIRED_SHADOW	Configuration scheduled for downlink but not yet confirmed by the device. New configurations can be created through MQTT, REST, my.sentinum.de, or the Sentinum registry UI. After successful application by the sensor, this value returns to an empty object: {}.
<BASETOPIC>/DEVICE_INFO	Additional device information that is too large for telemetry metadata, including deeper FOTA and connectivity/network information.

2.3 WRITE-ONLY TOPICS

Clients can publish to these topics. They cannot subscribe to them. Subscription attempts are denied and can cause the client to be disconnected.

TOPIC	DESCRIPTION
<BASETOPIC>/UPDATE_SHADOW	Publishes a new configuration that should be applied to the device. Valid configuration items appear in desired_shadow until the sensor applies them. After successful application, reported_shadow is updated. Use reported_shadow to infer the schema required for an update_shadow message.
<BASETOPIC>/GET_SHADOW	Triggers Datahub to publish the current reported_shadow, desired_shadow, and device_info again. The request payload is ignored and may be empty.

Shadow payloads use the base form `{"config": {}, "cmd": {}}`. Use config for configuration values and cmd for commands. Both sections are grouped by Sentos modules.

3. EXAMPLES

3.1 SUBSCRIBE TO TELEMETRY AND DEVICE STATE

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/up

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/reported_shadow

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/desired_shadow

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/device_info

3.2 TRIGGER A SHADOW REFRESH

Publish any payload to:

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/get_shadow

Example payload:

```
{
  "config": {},
  "cmd": {}
}
```

Datahub then publishes the
current reported_shadow, desired_shadow,and device_info on their respective read-only topics.

3.3 UPDATE A DEVICE CONFIGURATION

Publish the desired configuration delta to:

datahub/acme_corp/apollon/apollon/apollon_cellular_012345678901236/update_shadow

Example payload:

```
{
  "config": {
    "apollon": {
      "period": 60
    }
  },
  "cmd": {}
}
```

The change appears in desired_shadow until the device has received and applied it. After confirmation, reported_shadow contains the active value and desired_shadow returns to {}.