

HyCloud APIv1

This document describes the HyCloud API. The API is a typical JSON REST-like over HTTPS API. TLS is mandatory, the API is not exposed on plain HTTP.

API Revisions

- **19th Sep 2022:** Add the TA-Smart DP related errors.
- **12th Jul 2022:** Add the `error_status` property to data points.
- **25th May 2022:** Add documentation for Firmware API and mark calls that require Authentication.
- **12th May 2022:** The `Requester-ID` header is now required to make API calls.
- **14th Feb 2022:** Initial Release

Roadmap

- Add new properties that can be fetched in the data points:
 - current regime
 - current error
 - current control mode
 - current override status
 - stab_dp (integer, avg 15secs): is always 0 (see TA-Smart code on 10th Feb 2022)
- Get more information about a given TA-Smart configuration
- Get current connectivity for a given TA-Smart
- OAuth 2.0 authentication in addition/instead of the API Key mechanism.

Versioning

To ensure retro compatibility, the API is versioned through a prefix in the URL (e.g., `/api/v1/`). Changes within a version can only be additive: a new field in the returned object, a new optional parameter or a new endpoint are all possible within the same API version. However, any modification in the behavior or any deletion are *not* possible within a version.

Additions to a version will be notified in the [API Revision](#) section.

Optional & Nullable fields

When performing a request, an optional field can be omitted. In the case of a response, the field won't be listed at all in the object.

A nullable field on the other hand will always be there but might be set to `null`.

Parameters in Query String

The url should be url-encoded. Arrays can be passed as a query string with the following form: `/api/v1/whatever?values[]=42&values[]=46`. This will result in a array `values { 42, 46}`.

Errors

Errors are returned with a corresponding HTTP Status code depending on the type of error that was encountered.

The body of the response sticks as closely as possible to the [RFC7807](#). It consists of a JSON body with the following properties:

- **status** (integer): a repetition of the HTTP Status code,
- **type_uri** (string): a unique description of the issue,
- **title** (string): the title of the error,
- **details** (string, optional): a detailed explanation of the encountered issue.

The **status** and **type_uri** are stable throughout revisions of a given API version while the other properties *might* evolve with revisions. They should not be used by the API consumer to perform certain logic.

Example:

```
{
  "status" : 400,
  "type" : "api/v1/invalid-query",
  "title" : "Invalid Query",
  "details" : "Cannot parse the <to> date parameter. We expect a datetime
in the ISO 8601/RFC3339 format (e.g., '2022-02-
08T13:27:47.877745541+00:00')"
```

Error Types

APIv1 only exposes the following **type_uri**:

- "api/v1/invalid-query"
- "api/v1/not-found"
- "api/v1/unauthorized"
- "api/v1/server-error"

Authentication & Authorization

Requester-ID

All API calls must have a **Requester-ID** header. This field identifies the origin of an API call and is mandatory. In the future, this field might be used to throttle usage. You will be provided with a Requester-ID that identifies you. You should keep this secret as much as possible as it is use to measure your consumption.

The header should have the following format:

```
Requester-ID: <Your Requester ID>
```

Authentication

Some API calls must be done in the context of a user. This requires an authentication mechanism. The endpoint that requires authentication are explicitly marked as such.

Authentication is provided through an API Key passed in the request HTTP headers. This API key is tied to a user in HyCloud and can perform the same actions as the user would do through the HyCloud web interface.

A user can have different API Keys for different purposes. Even though those API Keys provide the same level of access, it might be a good idea to generate different keys for different contexts (e.g., CLI tool, regular server-to-server fetch).

Warning: Keep those API Keys tokens private and revoke any key that might have leaked to non-authorized parties.

The HTTP header containing the key should have the following format:

```
Authorization: ApiKey <API key>
```

To create an API Key, go to *Settings/API Keys* on HyCloud.

Base URLs

All endpoints are accessible on the main domains:

- production: <https://cloud.imi-hydronic.com/api/v1/>
- staging: <https://staging.cloud.imi-hydronic.com/api/v1/>

Beware that the two environments do not share any data. A user/api key from one environment has no link whatsoever with the other environment.

Endpoints

GET **/projects** (auth. required)

Returns the projects list of the current user has access to. This endpoint has no parameters or payload.

Response

JSON Array of **ProjectSummary** object with the properties:

- **id** (uuid): unique identifier of the project,
- **href** (string): the URL of the full project object,
- **name** (string): name of the project,
- **user_count** (integer): number of users having access to the project, no matter which access level they have,
- **tasmart_count** (integer): the number of TA-Smart in the project.

Example

/api/v1/projects

```
[
  {
    "href": "/api/v1/projects/2c6a8634-1d87-4d42-a61b-456ae04785c9",
    "id": "2c6a8634-1d87-4d42-a61b-456ae04785c9",
    "name": "IMI eRnD Office",
    "user_count": 3,
    "tasmart_count": 3
  },
  {
    "href": "/api/v1/projects/e2d83e26-9b34-449f-8a1e-c504e6962c59",
    "id": "e2d83e26-9b34-449f-8a1e-c504e6962c59",
    "name": "Creative Monkeys Mons",
    "user_count": 1,
    "tasmart_count": 0
  },
  {
    "href": "/api/v1/projects/c0385dcf-5bae-47d5-9b86-21b4f78b000e",
    "id": "c0385dcf-5bae-47d5-9b86-21b4f78b000e",
    "name": "IMI eR&D Lab",
    "user_count": 1,
    "tasmart_count": 2
  },
  {
    "href": "/api/v1/projects/7fa64c83-09bd-403a-9d96-45f17b309abc",
    "id": "7fa64c83-09bd-403a-9d96-45f17b309abc",
    "name": "Havr  Home Office",
    "user_count": 2,
    "tasmart_count": 0
  }
]
```

GET /api/v1/projects/<project_id> (auth. required)

Returns the details of a specific project.

Response

A Project JSON Object of the following form:

- **id** (uuid): unique identifier of the project,
- **name** (string): name of the project,
- **description** (string, nullable)
- **country** (string)
- **city** (string)
- **zipcode** (string, nullable),
- **address_line_1** (string, nullable)

- `address_line_2` (string, nullable)
- `lat` (float), latitude based on address geocoding,
- `lng` (float), longitude based on address geocoding,
- `users` (list of object), each object has:
 - `email` (string), the email of the user,
 - `access` (string): access level for this user. Either `read-only` or `admin`
- `tasmarts` (list of object):
 - `id` (uuid): General identifier of the TA-Smart,
 - `kind` (string): The type of TA-Smart (e.g., TA-Smart DN32)
 - `cloud_identifier` (string): the identifier of the TA-Smart for the cloud provider,
 - `full_sn` (string): full serial number of the TA-Smart (as indicated on the box),
 - `short_sn` (string): short serial number, makes it easier to quickly identify a TA-Smart,
 - `user_id`: (string, optional), optional identifier that can be set by the user. This identifier will be used when the TA-Smart is advertising in BLE,
 - `fw_version` (string, optional): version of the main firmware,
 - `webapp_version` (string, optional): version of the embedded webapp,
 - `ble_bootloader_version` (string, optional): bluetooth co-processor,
 - `ble_soft_device_version` (string, optional): bluetooth co-processor,
 - `ble_fw_version` (string, optional): bluetooth co-processor,
 - `href` (object): urls for endpoints related to the TA-Smart. Properties:
 - `datapoints` (string): base url for the datapoints endpoint.

Example

`/api/v1/projects/c0385dcf-5bae-47d5-9b86-21b4f78b000e`

```
{
  "id": "c0385dcf-5bae-47d5-9b86-21b4f78b000e",
  "name": "IMI eR&D Lab",
  "description": "Test ",
  "country": "Belgium",
  "city": "Mont-Saint-Guibert",
  "zipcode": "",
  "address_line_1": "",
  "address_line_2": "",
  "lat": 50.6365628,
  "lng": 4.6127019,
  "users": [
    {
      "email": "francois@tamere.eu",
      "access": "admin"
    }
  ],
  "tasmarts": [
    {
      "id": "f8bd23c5-bca9-472f-8a81-1d27229c40c1",
      "kind": "TA-Smart DN32",
      "full_sn": "00000000000000000000000000000000",
      "short_sn": "AAA",
```

```

    "cloud_identifier": "sv000000000000000000",
    "user_id": "SV-000000",
    "fw_version": "0.0.9.35",
    "webapp_version": "0.0.2",
    "ble_bootloader_version": "0.3",
    "ble_soft_device_version": "0.7.2.0",
    "ble_fw_version": "0.1.0.3",
    "href": {
      "datapoints": "/api/v1/ta-smart/f8bd23c5-bca9-472f-8a81-1d27229c40c1/data/points"
    }
  },
  {
    "id": "22b4133a-7ab6-40de-ae37-e41072c0d46c",
    "kind": "TA-Smart DN32",
    "full_sn": "39003C000851383408513834",
    "short_sn": "F8ED07AA",
    "cloud_identifier": "sv0123506284C73E02EE",
    "user_id": "TA-Smart 32 F8ED07AA",
    "fw_version": "0.0.9.74",
    "webapp_version": "2.0.0",
    "ble_bootloader_version": "255.255",
    "ble_soft_device_version": "0.7.0.1",
    "ble_fw_version": "0.0.0.18",
    "href": {
      "datapoints": "/api/v1/ta-smart/22b4133a-7ab6-40de-ae37-e41072c0d46c/data/points"
    }
  }
]
}

```

GET `/api/v1/ta-smart/<tasmart_id>/data/points` (auth. required)

This endpoint lets you retrieve data points sent by the TA-Smart. The valve sends it data with a resolution of *15secs*. Some values are averaged over those 15 secs (e.g., temperatures) while others are a snapshot at the time of capture (e.g., actuator position)

In case of power or connectivity issue, it is possible to have *holes* in the data points.

All units are expressed in a way to avoid floating point numbers. Convert them accordingly.

Query String Params:

- **from** (string, optional, default: `1 hour ago`): lower time bound. Datetime formatted according to ISO 8601/RFC3339,
- **to** (string, optional, default: `now()`): upper time bound. Datetime formatted according to ISO 8601/RFC3339,
- **limit** (integer, optional, default: `240`, max: `2000`): 240 data points covers ~1h of continuous data,
- **offset** (integer, optional, default: `0`):

- **direction** (integer, optional, default: **descending**). The ordering of the points based on their time property. Possible values:
 - **descending** (default)
 - **ascending**
- **properties** (list of string, optional, default: **empty list**). The properties of the datapoint to fetch. Possible values:
 - **valve_timestamp**: the timestamp as the TA-Smart knows it. If the clock is not synchronized, it might drift from the **time** reported in the response. The timestamp is the number of seconds since the 1st January 2000 (à la Zigbee timestamp),
 - **valve_dst**: the Daylight Saving time switch of the TA-Smart,
 - **measured_flow** (integer, avg 15secs): the measured flow (*dl/h*),
 - **relative_flow** (integer, avg 15secs): the relative flow of the current maximum (*1/100%*),
 - **measured_power** (integer, avg 15secs): the measured power (*dW*),
 - **relative_power** (integer, avg 15secs): the power relative to the current max (*1/100 %*),
 - **energy_counter_regime_1** (integer): the total energy on regime 1 (*kWh*),
 - **energy_counter_regime_2** (integer): the total energy on regime 2 (*kWh*),
 - **position** (integer): the current relative position of TA-Slider controlled by the TA-Smart (*1/100% of current max*),
 - **input_signal** (integer): the value of the input signal (*dV*),
 - **local_temp** (integer, avg 15secs): the temperature sensor value (*dC°*),
 - **remote_temp** (integer, avg 15secs): the remote temperature sensor value (*dC°*),
 - **error_status**: the current error status of the TA-Smart at the given time

Response

An **DatapointQueryResult** object containing the data and summarizing the options that were used to perform the query:

- **from** (datetime),
- **to** (datetime),
- **sort** (string): 'ascending' or 'descending',
- **limit** (integer),
- **offset** (offset),
- **properties** (list of string), the properties that were fetched,
- **points**: (list of datapoints), object containing the points. Each object will contain the requested properties and an additional **time** field containing the ingestion date of the datapoint.

Examples

```
GET /api/v1/ta-smart/c8c2f9a6-abff-4584-bf1d-7b28daa034db/data/points
?from=2022-02-08T12%3A27%3A47.877745541%2B00%3A00
&to=2022-02-10T13%3A27%3A47.877745541%2B00%3A00
&properties[]=measured_flow
&properties[]=measured_power
&limit=3
```

```
{
  "from": "2022-02-08T12:27:47.877745541+00:00",
  "to": "2022-02-10T13:27:47.877745541+00:00",
  "sort": "descending",
  "limit": 3,
  "offset": 0,
  "properties": [
    "measured_flow",
    "measured_power"
  ],
  "points": [
    {
      "time": "2022-02-09T14:58:24.535091Z",
      "measured_flow": 15340,
      "measured_power": 151814
    },
    {
      "time": "2022-02-09T14:58:09.599091Z",
      "measured_flow": 15346,
      "measured_power": 152099
    },
    {
      "time": "2022-02-09T14:57:54.542091Z",
      "measured_flow": 15334,
      "measured_power": 152275
    }
  ]
}
```

GET /api/v1/firmwares/ta-smart/latest

Get the latest firmware for the given Requester-ID. Callers can also add a header to tell which version of the App is making the call (eg., `App-Build-Number: 130`).

Response

A Firmware Bundle response with the following properties:

- `download_link`: a link to download the firmware bundle file. Beware that this link expires at some point.
- `metadata`: an object describing the versions from the bundle

Examples

```
GET /api/v1/firmwares/ta-smart/latest
```



```
{
  "download_link": "/firmwares/c375a179-8a7f-4c21-b877-458ee94a3dbe.zip",
  "metadata": {
    "published_at": "2022-05-16T12:43:11.634082Z",
    "bt840_bootloader": "0.3",
    "bt840_fw": "0.1.0.9",
    "bt840_softdevice": "7.2.0",
    "fw": "0.1.0.9",
    "webapp": "2.0.0",
    "updater": "1.0.0-alpha.7"
  }
}
```

Appendix

Entities as Rust Code

Those entities used by the API are here for reference.

```
#[derive(Debug, Serialize, Deserialize)]
pub struct ProjectSummary {
    pub href: String,
    pub id: Uuid,
    pub name: String,
    pub user_count: i64,
    pub tasmart_count: i64,
}

#[derive(Debug, Serialize, Deserialize)]
pub enum ProjectAccessLevel {
    #[serde(rename = "read-only")]
    ReadOnly,

    #[serde(rename = "admin")]
    Admin,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct Project {
    pub id: Uuid,
    pub name: String,
    pub description: Option<String>,
    pub country: String,
    pub city: String,
    pub zipcode: Option<String>,
    pub address_line_1: Option<String>,
    pub address_line_2: Option<String>,
    pub lat: f64,
    pub lng: f64,
    pub users: Vec<User>,
}
```

```

    pub tasmarts: Vec<TASmartSummary>,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct User {
    pub email: String,
    pub access: ProjectAccessLevel,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct TASmartSummary {
    pub id: Uuid,
    pub kind: String,
    pub full_sn: String,
    pub short_sn: String,
    pub cloud_identifier: String,
    pub user_id: Option<String>,
    pub fw_version: Option<String>,
    pub webapp_version: Option<String>,
    pub ble_bootloader_version: Option<String>,
    pub ble_soft_device_version: Option<String>,
    pub ble_fw_version: Option<String>,
    pub href: TASmartHref,
}

#[derive(Debug, Serialize, Deserialize)]
pub enum TASmartError {
    ErrorLowPower,
    ErrorInputLineBroken,
    WarningFlowNotReached,
    WarningPowerNotReached,
    ErrorLocalTempSensorDisconnected,
    ErrorRemoteTempSensorDisconnected,
    ErrorLocalTempSensorShortCircuit,
    ErrorRemoteTempSensorShortCircuit,
    ErrorLocalTempSensorBelowMin,
    ErrorRemoteTempSensorBelowMin,
    ErrorLocalTempSensorAboveMax,
    ErrorRemoteTempSensorAboveMax,
    WarningActuatorManualOverride,
    ErrorActuatorBlocked,
    ErrorFlowMeasurement,
    WarningAirBubbles,
    ErrorReverseFlow,
    ErrorActuatorComFailure,
    ErrorDpSensorDisconnected,
    WarningAvailDpTooLowForDpStab,
    WarningKvLoadTooHighForDpStab,
    WarningDpStabAboveMax,
    Other,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct TASmartHref {

```

```
    pub datapoints: String,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct DatapointQueryResult {
    pub from: String,
    pub to: String,
    pub sort: SortDirection,
    pub limit: i64,
    pub offset: i64,
    pub properties: Vec<DataProperty>,
    pub points: Vec<Datapoint>,
}

#[derive(Debug, Serialize, Deserialize)]
pub struct Datapoint {
    pub time: DateTime<Utc>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub valve_timestamp: Option<i64>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub valve_dst: Option<i16>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub measured_flow: Option<i32>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub relative_flow: Option<i32>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub measured_power: Option<i32>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub relative_power: Option<i32>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub local_temp: Option<i16>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub remote_temp: Option<i16>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub input_signal: Option<i16>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub actuator_position: Option<i16>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub energy_counter_regime_1: Option<i64>,

    #[serde(skip_serializing_if = "Option::is_none")]
    pub energy_counter_regime_2: Option<i64>,
}
```

```
#[serde(skip_serializing_if = "Option::is_none")]
pub error_status: Option<Vec<TASmartError>>,
}

#[derive(Debug, Clone, Copy, Serialize, Deserialize, FromFormField,
PartialEq)]
#[serde(rename_all = "lowercase")]
pub enum SortDirection {
    Ascending,
    Descending,
}

#[derive(Serialize, Deserialize, PartialEq, Clone, Copy, Debug)]
pub enum DataProperty {
    #[serde(rename = "valve_timestamp")]
    ValveTimestamp,

    #[serde(rename = "valve_dst")]
    ValveDST,

    #[serde(rename = "measured_flow")]
    AvgMeasuredFlow,

    #[serde(rename = "relative_flow")]
    AvgRelativeFlow,

    #[serde(rename = "measured_power")]
    AvgMeasuredPower,

    #[serde(rename = "relative_power")]
    AvgRelativePower,

    #[serde(rename = "energy_counter_regime_1")]
    EnergyCounterRegime1,

    #[serde(rename = "energy_counter_regime_2")]
    EnergyCounterRegime2,

    #[serde(rename = "position")]
    ActuatorPosition,

    #[serde(rename = "input_signal")]
    InputSignal,

    #[serde(rename = "local_temp")]
    AvgLocalTemperature,

    #[serde(rename = "remote_temp")]
    AvgRemoteTemperature,

    #[serde(rename = "error_status")]
    ErrorStatus,
}
```

```
/// We keep the metadata as a separate object so the mobile apps can
/// save the whole object without the download link.
#[derive(Serialize, Deserialize)]
pub struct FwBundleMetadata {
    pub published_at: DateTime<Utc>,
    pub bt840_bootloader: String,
    pub bt840_fw: String,
    pub bt840_softdevice: String,
    pub fw: String,
    pub webapp: String,
    pub updater: String,
}

#[derive(Serialize, Deserialize)]
pub struct FwBundle {
    pub download_link: String,
    pub metadata: FwBundleMetadata,
}
```