

Zednet REST API

v1.1 2018-01-10

Zednet exposes a REST style API to allow third party applications access to data.

API request format:

In general the API follows the format:

GET <https://live.zednet.co.za/index.php/api/1/channels/1?key=12345>

In this instance a GET request is made for a Zednet Channel with ID=1, and submits API key 12345 for authentication.

The general format is

/channels - for all of the channels on user account

/channels/1 - for information just on channel 1

/channels/1,2,3 for information on 3 specified channels

In addition to the API key, some API endpoints also support filters/flags submitted as query string parameters:

[/channel_data?format=csv&data_type=4](#)

API Authentication:

1. Permanent API key

To remain compatible with legacy clients, the API implements a simple, permanent API key.

The key uniquely authenticates to a particular Zednet user, and is set up in the Zednet admin USERS interface:

Name	Value
ID	482
Client	ZEDNET VISITOR
Usergroup	clientuser
Username	visitor-apiuser
Password	*****
Email	rean@thewoodshed.co.za
Active	yes
Api Key	APIKEYHERE
Name	Api User
Designation	
Department	

Given the relatively low security of a permanent key (even on HTTPS), it is considered good practice to set up a key on a “client user” level login - i.e. a dedicated API user, with read only access to a Zednet client account.

The API key can be submitted in one of two ways:

1. (preferred) set header Authorization = key
2. Add key as query string parameter as in example above /channels/1?key=12345

2. JSON Web Token (JWT)

The preferred method of authentication is to use a JSON web token, or JWT. This does not require any setup in the back end, and keys (tokens) are generated on request: Your application makes a POST call to the API with username and password of an existing Zednet user. The JWT in this response is then sent with further requests:

POST <https://live.zednet.co.za/index.php/api/1/token>

using x-www-form-urlencoded

with username and password fields in the body

This returns a JWT token which is valid for 60 minutes, or 404 error response if login invalid.

When the token expires, you can either login again, or just before expiry call

PUT <https://live.zednet.co.za/index.php/api/1/token>

with Authorization: Bearer XXXXXXXX in header

which will issue a new token with fresh expiry. This then needs to be used in future requests.

3. Authentication “scope”

Regardless which mechanism is used, login with “client” or “client-admin” level permission are automatically scoped to their client account. In other words, a simple call to get “all sites” will return only sites belonging to the authenticated client account.

If information is required on multiple client accounts, a full “admin” level login can be used. Queries can then be scoped using ID or filters as the particular endpoint allows.

JWT authentication offers one further useful mechanism:

With a single login, full admins can change the scope of queries to particular client account by making a call to

PUT <https://live.zednet.co.za/index.php/api/1/token>

while adding a field: `client_id = 16` to the body of the put request.

This will issue a new token with fresh expiry and scoped to `client_id 16` - which then needs to be used in further requests.

Using this token, further access is automatically scoped to that client ID.

So all queries will return only sites/channels data etc from the one client account without having to specify filters.

Admins can move between client accounts by making a fresh PUT call for re-scoped JWT.

API response format:

Raw timeseries data can be returned in multiple formats (documented below), incl CSV.

All other responses are either CSV or JSON. JSON is the default, but CSV can be requested by setting the ‘Accept’ header to `text/csv`. In general JSON is recommended for everything but raw time series data as it provides the required structure to make sense of responses.

a JSON response is encoded as follows:

A failed request will be returned with appropriate HTTP error code (404, 403 etc) and JSON payload identifying problem:

```
{
```

```
"success": false,  
"message": "You do not have permission to view channel(s) 3"  
}
```

Successful request will be returned with HTTP 200 response code and payload similar to:

```
{  
  "success": true,  
  "records": [  
    {  
      "channel_id": "318",  
      "name": "AZP Pressure",  
      etc.  
    },  
    {  
      etc..  
    }  
  ]  
}
```

API endpoints:

To start off, a brief note on Zednet hierarchy or database model:

1. sources

1. Raw logger data is stored in Zednet as “sources” - a source might be manual csv upload, or a stream of data on particular logger channel.
2. A source has an internal ID in Zednet, always unique to a zednet instance.
3. A source also has an “external ID” - this maps the source record to its external data on import.
 1. It is typically composed as follows:
 - prefix_logger_channel
 - pmac1_1234_01
 - pmac1_i_1234_01 (for index)
 2. Prefix identifies the inbox, and ensures that clashing PMAC ID’s can import (ID 1234 may e.g. be used on PMAC 1 but also PMAC 2 etc)
 3. In the case of Technology data, this means you e.g. need to know the inbox prefix (or PMAC instance) as well as PMAC ID and channel to guess at external ID.

4. One channel can have one or more sources which are automatically appended when data is queried. Typically multiple sources will cover different time range, only latest source should be receiving data. This allows the logical construct of a “measurement” (ie pressure at point A) to remain constant, while the underlying hardware supplying data can change.
 5. A channel might also be derived (from other channels) and so have no direct source.
 6. You should never query data from source directly (but rather on channel)
 1. because the channel may apply adjustments, and contains the “business logic” to converts raw data to a measurement at a point.
2. channels
 1. e.g. the pressure/flow at point A.
 2. one or more sources combine to provide a logical data channel
 3. the source behind a channel may be changed (e.g. because logger is replaced)
 4. channel data may be adjusted from raw data (e.g. scale factor applied)
 5. multiple channels might be combined to form a “virtual” channel - e.g. combined flow
 3. site
 1. all channels are associated with a site, a physical location - i.e. point A
 2. a site can optionally have a geographic location
 4. geo_group
 1. all sites in turn are organised into geographical groups
 2. geo groups are hierarchical
 1. there might e.g. be regions, districts and zones (or any other grouping a client sets up)
 3. geo groups can optionally be supplied with spatial co-ordinates
 5. client
 1. All geo_groups, sites, channels belong to a particular client
 2. Users also belong to a client - so when they log in, all their interactions are automatically scoped to the client. They can only see the sites, channels etc that belong to the same client that they are registered with.
 3. Full admin users belong to a client, but they can access information across all clients in a Zednet instance.
 6. types
 1. geo_groups, sites, channels can optionally have a “type”.
 2. So geo_groups might e.g. be regions/districts/zones while sites can be reservoirs/PRV chambers/amr etc
 3. types are created dynamically - they are not fixed across Zednet instances

API endpoints that are presently implemented:

GET <https://live.zednet.co.za/index.php/api/1/sources>

For all sources that an API key has access to
(Note this returns only the sources linked to the API key user’s channels)

GET <https://live.zednet.co.za/index.php/api/1/sources/12>

For a particular source
(the API key must have access to the requested source)

It accepts the following query string parameters (in addition to the API key)

- `extid=pmac1_1234_01`
 - Retrieve a source on its external ID
- `channel=13`
 - Retrieve sources linked to a channel

GET https://live.zednet.co.za/index.php/api/1/channel_data

This returns raw time series data on one or more logger channels.

It accepts the following query string parameters (in addition to the API key)

- `start=2016-01-01 10:00:00`
 - note start and end dates are included in returned set (i.e. from start, including end)
- `end=2016-01-02`
 - where time is not included, 00:00:00 is assumed for start, and 23:59:59 for end
- `days=2`
 - where start and end dates are supplied, days setting is ignored
 - if no start/end is supplied, `end=now()` and `start = end - days`
 - if only start or end is supplied, days are used to calculate offset
- `channels=1,2,3`
 - one or more channel ID's to return data on
 - if none supplied, all client channels are included
- `data_types=1,2,3`
 - filter on one or more data types by data type ID
- `format=csv (csv/shef/hydstra)`
 - If no format is supplied, CSV is returned with one row header
- `reference=1 (1/0)`
 - If `reference=1` (the default) then data is exported using the channel "reference" field as key if it is available.
 - This makes it possible for clients to re-map the Zednet ID system to their own before export

GET https://live.zednet.co.za/index.php/api/1/channel_data_summary

This returns summary stats for one or more channels. The summary includes latest date, latest value/meter as well as avg/min/max/sum as relevant. By default the summary is for most recent 7 days data on record, but query filters can be used to change date/duration

- `start=2016-01-01 10:00:00`
 - note start and end dates are included in returned set (i.e. from start, including end)
- `end=2016-01-02`
 - where time is not included, 00:00:00 is assumed for start, and 23:59:59 for end
- `days=2`
 - where start and end dates are supplied, days setting is ignored

- if no start/end is supplied, end=now() and start = end - days
 - if only start or end is supplied, days are used to calculate offset
- channels=1,2,3
 - one or more channel ID's to return data on
 - At least one channel_id must be supplied.

GET <https://live.zednet.co.za/index.php/api/1/channels/1,2,3>

This returns meta data (name, site id, etc) on one or more logger channels.

The last segment of the URL (optionally) identifies one or more channels by ID - omitting this returns all channels.

It accepts the following query string parameters (in addition to the API key)

- site=1,2,3
 - one or more site ID's to filter on
- shared=1 (or 0)
 - only relevant where no ID's have been supplied - must the response include channels that have been "shared" by other clients.

GET <https://live.zednet.co.za/index.php/api/1/sites/1,2,3>

This returns meta data (name, group id, etc) on one or more physical sites. Channels are all linked to site

The last segment of the URL (optionally) identifies one or more sites by ID - omitting this returns all sites.

It accepts the following query string parameters (in addition to the API key)

- shared=1 (or 0)
 - only relevant where no ID's have been supplied - must the response include sites that have been "shared" by other clients.

GET https://live.zednet.co.za/index.php/api/1/geo_groups

This returns meta data (name, parent id, etc) on one or more geo groups. Sites are all linked to a geo group.

This endpoint returns all geo_groups associated with a particular client account.

GET <https://live.zednet.co.za/index.php/api/1/clients>

This returns meta data (name, id, etc) for all the clients that a user account has access to. Typically used by admin level JWT login to be able to alter scope to one or more client accounts.

GET <https://live.zednet.co.za/index.php/api/1/types/1,2,3>

This returns either some or all the “types” of channel,site, geo group that have been set up on the system.

It accepts the following query string parameters (in addition to the API key)

- `for=all` (or `sites/channels/assets/geo_groups`)
 - limit the returned types to just site types, or channel types

GET https://live.zednet.co.za/index.php/api/1/data_types/1,2,3

This returns either some or all the physical data types configured on the Zednet instance. Each data type has a name as well as unit that can be used on charts etc.