

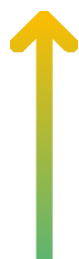


GDR API

Implementation guide

Version history

Version	Data	Change
1.0	04-02-2025	Initial Version



Contents

Getting started	4
Introduction	4
Developers Portal	4
Interactive API-Specification	4
Timestamps	7
API Keys	7
GDR Requests	8
Intended Use	8
Implementation	8
Get Device	8
Get Channel(s)	9
Sending data	9
Deprecated endpoints	10



Getting started

Introduction

The Generic Data Receiver (GDR) API of Fudura allows you, as a user, to submit measurement data from configured devices to Fudura.

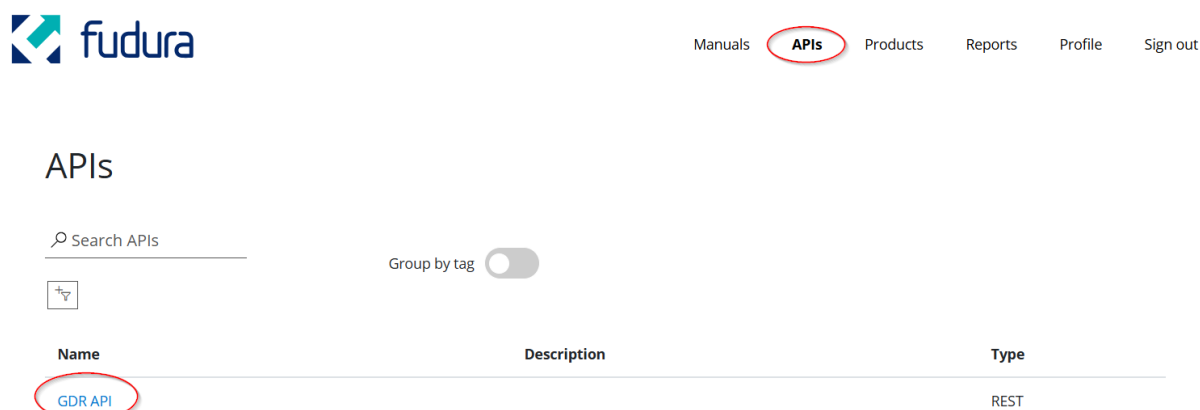
This document explains how to utilize the GDR API.

Developers Portal

To get started you need an account for Fudura's Developer Portal (<https://developers.fudura.nl>). This account can be obtained via Sales / Customer Support. An invite to create an account will be sent to the provided e-mail address. After creating a password and logging in to your newly created account you can access the specification for the GDR API, retrieve API keys and perform test requests against the API.

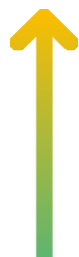
Interactive API-Specification

The specification for the GDR API can be found under "APIs" "GDR API".

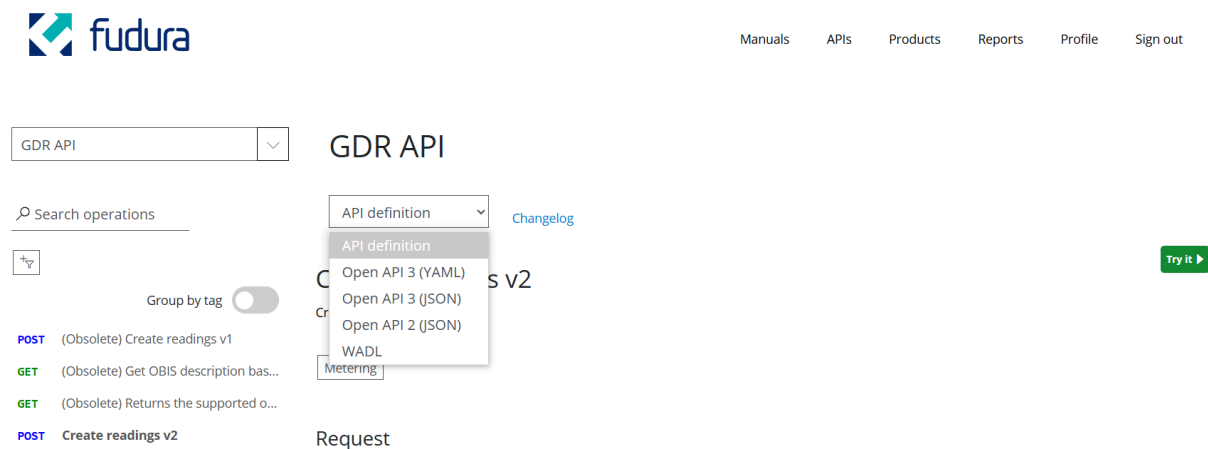


The screenshot shows the Fudura Developers Portal interface. At the top, the Fudura logo is on the left, and navigation links for 'Manuals', 'APIs', 'Products', 'Reports', 'Profile', and 'Sign out' are on the right. The 'APIs' link is circled in red. Below the navigation bar, the page title 'APIs' is displayed. There is a search bar labeled 'Search APIs' and a 'Group by tag' toggle switch. A table lists the available APIs. The first entry, 'GDR API', is circled in red. The table has three columns: 'Name', 'Description', and 'Type'.

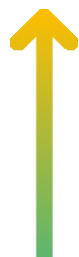
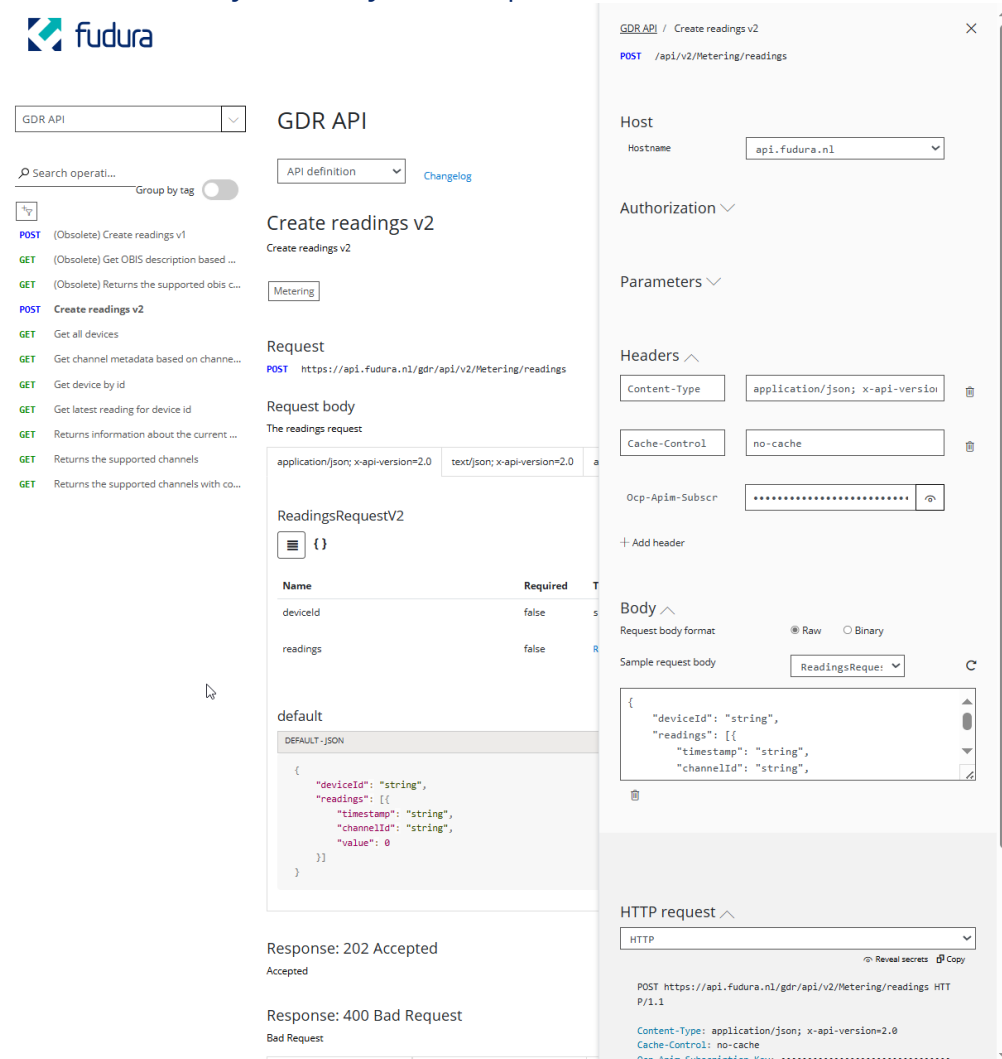
Name	Description	Type
GDR API		REST



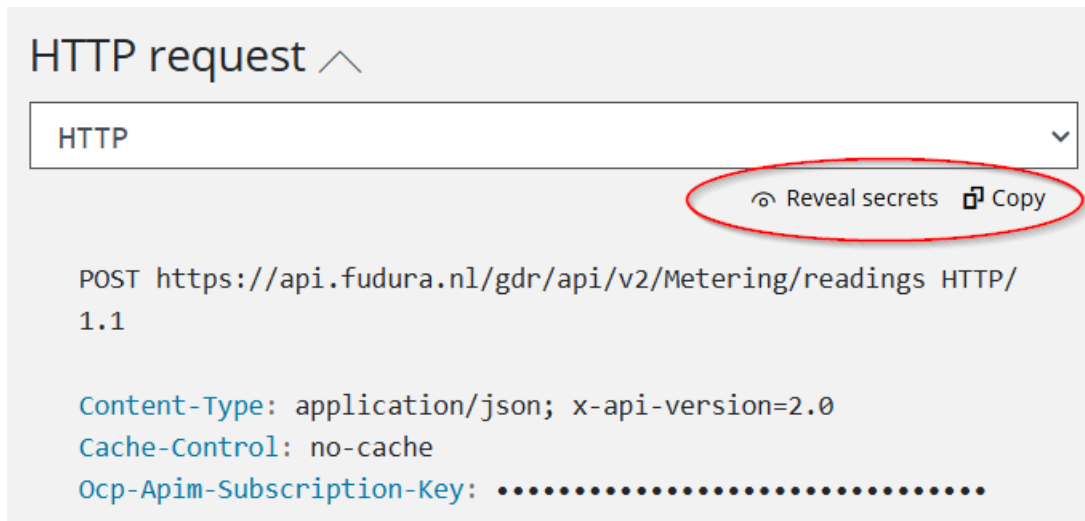
The specification is provided in both an interactive, graphical format as well as a downloadable document by selecting the desired format in the dropdown:



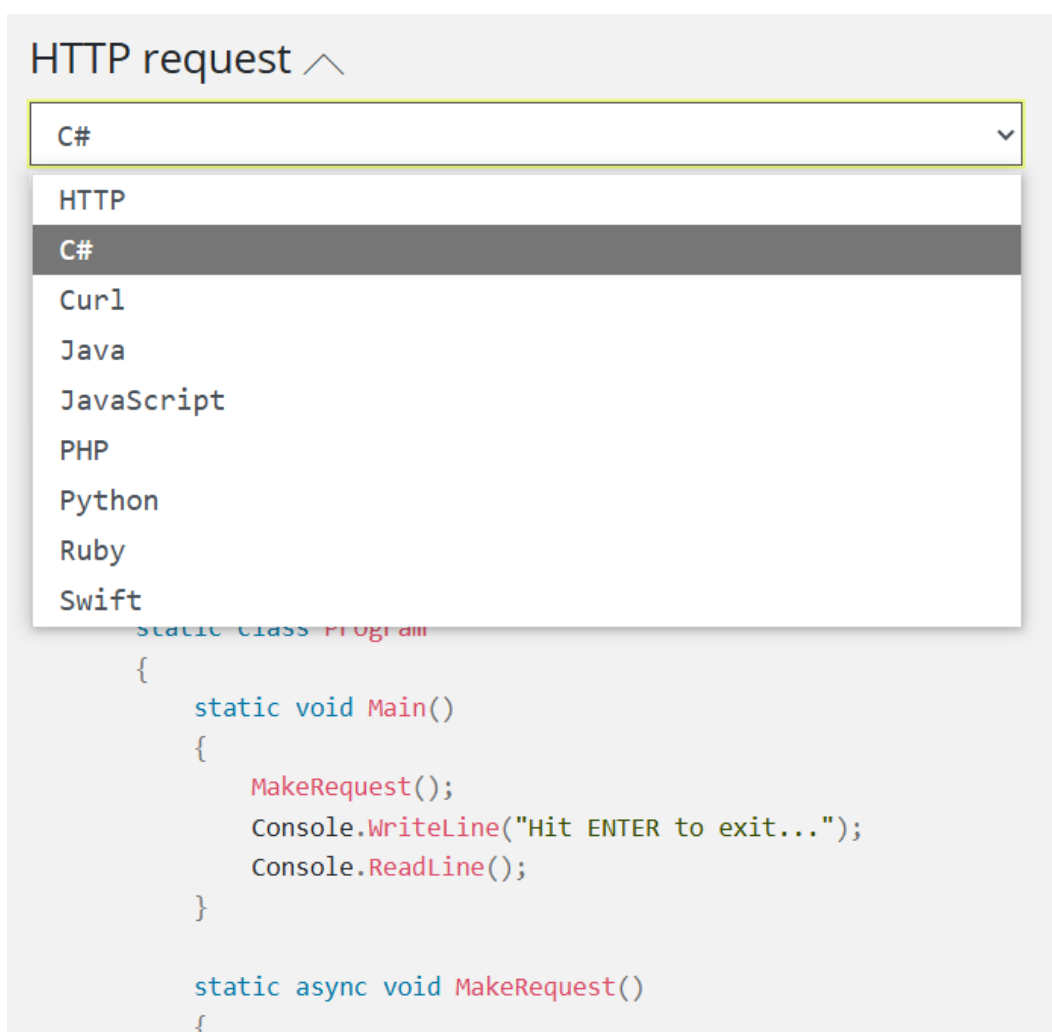
The graphical interface allows you not only to browse the endpoints, but also to send requests directly from the browser. The “try it” button (on the right) opens a new interface where you can try out a request.



After sending the request you get the options to reveal the secrets, or to copy the full request and used headers with just one click.



And by opening the dropdown at the HTTP request you can get examples on how to implement the request in code for the most popular programming languages.



Timestamps

All timestamps used in the GDR API are in UTC, in the “Zulu-time”-format. For each consumption reading the ReadingTimestamp represents the until-timestamp. For example: for the electricity consumption from 2023-07-25T15:45Z till 2023-07-25T16:00Z the ReadingTimestamp will be 2023-07-25T16:00Z.

API Keys

Your API Keys can be found under “Profile” under “Subscriptions”. You can view and regenerate the keys here. **Note: Keep these keys in a safe place! Do not commit a key into version control or share it with others. If you suspect a key might be leaked, regenerate it, and update the key in your application.**

User profile

Account details

Email

First name

Last name

Registration date

Change name

Change password

Close account

Subscriptions

Subscription details		Product	State	Action
Name		GDR API	Active	Cancel
Started on	05/31/2024			
Primary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx			Show Regenerate
Secondary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx			Show Regenerate



GDR Requests

Intended Use

This chapter shows how to implement an integration with the GDR API to provide Measuring data to Fudura. **Note: keep your client / code up to date with changes of the GDR API as obsolete endpoints will be removed after a certain amount of time and notifications, see chapter Deprecated Endpoints for changes and alternative endpoints.**

Implementation

To get started you need to understand these key concepts:

- **Metering Point Id** An identifier for a metering point, consisting of an EAN optionally followed by a suffix.
- **Channel Id** An identifier for a channel.

Providing measuring data is done via the `/gdr/api/v2/Metering/readings` endpoint, but before you can use this endpoint, you need to decide for which device and which channels you want to provide data. The process of getting this information and finally providing measurement data consists of the following steps:

Get Device

The first step is to query the available devices, for which reading data can be provided. This is done using the `gdr/api/v1/Devices` endpoint, listed as operation *Get All Devices* in the developer portal. This endpoint will provide a list of devices with the following properties:

- `deviceId`
- `meteringPointId`
- `medium` (For example: Generic, Electricity, HeatCost, Cooling, Heating, Gas, Water, HotWater)
- `customerReference`

Note: For adding or updating devices please contact support.



Get Channel(s)

With the results from step 1 you have a list of devices for which you can provide measurement data. But before you can provide this data you need to find the correct channels to provide data on. This is done using the `/gdr/api/v2/Channels` endpoint, listed as operation *Returns the supported channels*. This endpoint will provide a list of devices with the following properties:

- channelId
- description
- interval
- unitOfMeasurement

Alternatively you can use `/gdr/api/v2/Channels/Medium/{medium}` endpoint, listed as operation *Returns the supported channels with corresponding medium*. This endpoint returns the same properties but only returns channels which can be used for the corresponding medium. (this medium can be found in the result of get all devices request)

Sending data

Now you can send measurement data using the `/gdr/api/v2/Metering/readings` endpoint and the results of Get Device(s) and Get Channel(s). From the get devices result, you need to select a single deviceId per request. And from the get channels result you need to select one or more channels to provide data on. The channels must match reading data as channel defines the interval and unitOfMeasurement of the reading value.

Note: Make sure that readings timestamps match the channel interval of corresponding channel. For example, for the interval "00:15:00", data can be provided for timestamps "2025-01-29T13:00:00", "2025-01-29T13:15:00", "2025-01-29T13:30:00" and "2025-01-29T13:45:00", whereas the timestamp of "2025-01-29T13:00:01" would be invalid.

Note: The GDR API has a limit of 1000 calls per hour per use. To minimize the chance of reaching this limit and for efficiency it is preferred to send multiple readings (with a maximum of 1000) in a single request.

If the received measurement data is correctly formatted you will receive a 202 accepted response. Alternatively, you can check whether the measurement data is processed using the `gdr/api/v1/Metering/{deviceId}/latest` endpoint, listed as operation *Get latest reading for device id*. This endpoint will return the timestamp of the latest received measurement data. In case earlier readings have been provided later, then still the latest reading timestamp is returned. For example: Request 1: Timestamp: 2024-02-01 00:00:00 Request 2: Timestamp: 2024-01-15 00:00:00 Will return the timestamp of 2024-02-01 00:00:00



Deprecated endpoints

Some endpoints are deprecated, but still available for backwards compatibility. Use the suggested alternatives to make sure your application still works with upcoming versions.

Deprecated endpoint	Alternative
Create readings v1 (OBIS-code based)	Create readings v2 (Channel based)
Get supported obis codes	Get supported channels / Get supported channels by medium
Get OBIS description	Get channel metadata

