



ibaHD-Server-API-Read

gRPC-API interface to query historical data and events by own applications

Manual
Issue 1.1

Manufacturer

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts

Main office +49 911 97282-0
Fax +49 911 97282-33
Support +49 911 97282-14
Engineering +49 911 97282-13
E-mail iba@iba-ag.com
Web www.iba-ag.com

Unless explicitly stated to the contrary, it is not permitted to pass on or copy this document, nor to make use of its contents or disclose its contents. Infringements are liable for compensation.

© iba AG 2021, All rights reserved.

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site www.iba-ag.com.

Version	Date	Revision - Chapter / Page	Author	Version SW
1.1	02-2021	Web link for supported languages	GS	2.5.0

Windows® is a brand and registered trademark of Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.

Content

1	About this manual.....	4
1.1	Target group and previous knowledge	4
1.2	Notations	4
1.3	Used symbols.....	5
2	Introduction.....	6
2.1	gRPC.....	6
2.2	Programmatic usage of ibaHD-API.....	6
2.3	Testing the API	7
2.4	Transport security	7
2.5	Authentication	7
2.6	Design principles.....	7
2.7	Licenses	8
2.8	Memory management.....	8
3	ibaHD-API in ibaHD manager.....	9
3.1	General information	9
3.2	Configuration	9
3.2.1	Server	10
3.2.2	API keys	10
3.2.3	Certificates.....	11
3.2.4	Generate a new certificate	12
3.3	Get API key	12
4	ibaHD-API Functionality	13
4.1	GetHdStores()	13
4.2	GetHdStoreSchema()	13
4.3	GetRawChannelData()	13
4.4	GetAggregatedChannelData().....	15
4.5	GetEventData()	15
5	Sample Clients	16
6	Support and contact.....	17

1 About this manual

This manual describes the usage of the programming interface *ibaHD-Server-API-Read*. It has to be considered as documentation being complementary to the *ibaHD-Server* manual. If information is required we refer to the *ibaHD-Server* manual.

The latest version of this manual can be found on our website www.iba-ag.com under download.

1.1 Target group and previous knowledge

This documentation addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as a professional if he/she is capable of assessing the work assigned to him/her and recognizing possible risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons, who are concerned with capturing and storage of measuring data. For the handling of *ibaHD-Server-API-Read* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge *ibaHD-Server*
- Basic knowledge programming languages, especially C# or C++ or Python or another language which is provided by the gRPC group

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu <i>Logic diagram</i>
Calling the menu command	<i>Step 1 – Step 2 – Step 3 – Step x</i> Example: Select the menu <i>Logic diagram - Add - New function block</i> .
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
File names, paths	"Filename", "Path" Example: "Test.doc"

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!



The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.
-

Warning!



The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.
-

Caution!



The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures
-

Note



A note specifies special requirements or actions to be observed.

Tip



Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation



Reference to additional documentation or further reading.

2 Introduction

ibaHD-Server-API-Read is a licensed API enabling 3rd party applications to query historical data and events from *ibaHD-Server*.

The API is built on the open source gRPC framework, which is a Remote Procedure Call (RPC) framework based on the HTTP2 protocol and Protocol Buffers serialization.

For further information see <https://grpc.io/>. This website provides different tools and guides for the generation of gRPC-client code for various programming languages.

Version 1 (v1) of the API allows the client to retrieve information about the available data stores, the signal structure as well as channel and event data from *ibaHD-Server*.

In the following, *ibaHD-Server-API-Read* is referred to with the short name *ibaHD-API*.

2.1 gRPC

gRPC uses Protocol Buffers as IDL (Interface Definition Language) to define the API structure and message formats (<https://grpc.io/docs/guides/concepts/>).

The interface definition makes it possible to generate the client code for various programming languages. A complete list of currently supported languages can be found here:

<https://grpc.io/docs/languages/>

Once the client code has been generated it can be used to connect to the server-side of the API, which is built into *ibaHD-Server*.

2.2 Programmatic usage of ibaHD-API

To use the API programmatically, first the *ibaHD-API* client code for the desired programming language must be generated. To do this, two parts are required:

- The *ibaHD-API.proto* interface definition file. It is supplied with the *ibaHD-Server* installation and can be found in the “ibaHD-API” folder inside the *ibaHD-Server* installation directory.
- The gRPC tooling required to generate the *ibaHD-API* client. For each supported language the gRPC website provides a detailed tutorial how the client code can be generated:
<https://grpc.io/docs/tutorials/>

Once the *ibaHD-API* client code has been generated, it can be linked to the target application. Depending on the used language, it may be necessary to link additional gRPC and ProtoBuf base packages to the application. Details are provided in the programming language-specific guides.

Note



The gRPC toolchain for generating the client code is freely available as open source and not intellectual property of iba AG. The tools are subject to change without notice. It is recommended to make a backup of the code generation toolchain that was used to generate the client code for *ibaHD-API*.

2.3 Testing the API

There are multiple GUI applications available which allow loading gRPC interface definition files (.proto) and executing requests, for example BloomRPC <https://github.com/uw-labs/bloomrpc>, to test the API without generating and integrating the client code.

A selection of similar tools can be found here:

<https://github.com/grpc-ecosystem/awesome-grpc#tools-gui>

For the conversion of Unix time stamps in the different requests and responses, several tools can be found on the internet. One example is <https://currentmillis.com> which can be used for millisecond conversions.

Note



All these tools are freely available as open source and not intellectual property of iba AG. They are subject to change without notice. Function and availability cannot be guaranteed.

2.4 Transport security

Transport security over network is done via TLS certificates. As TLS is a mandatory requirement for *ibaHD-API* to function, certificates must be configured before the API can be enabled. Clients must reference the server TLS certificate to be able to connect to the API endpoint. See chapter [↗ Configuration](#), page 9 for information about certificate generation and configuration.

2.5 Authentication

Authentication is done on a per-user basis if user management is enabled. To authenticate a user, an API key must be generated for the user (see chapter [↗ Configuration](#), page 9). The API key must then be supplied with each request as a header field (also called metadata in gRPC context) named “ibahdapi-apikey”.

2.6 Design principles

The goal of *ibaHD-API* is to provide generic access to the data stored in *ibaHD-Server*. This generic approach allows to support a wide range of customer use cases.

Minor version changes of the API are always compatible to each other and should not require to regenerate the client code, unless new API features should be used.

Major version changes signal breaking changes in the API that make the regeneration of the client code mandatory and may require implementation changes.

2.7 Licenses

ibaHD-Server-API-Read is an additional licensed interface of *ibaHD-Server*. The license must be added to the dongle that contains the base license of *ibaHD-Server*. For further information please contact your local iba support.

2.8 Memory management

All data APIs have options to either limit or specify the amount of data that should be sent in a single response message. While there are default values in place, these limits may not be optimal for all use cases and should be adjusted for the target environment. The memory requirement for a single response message depends on a multitude of factors, such as original sample rate, requested time range, requested number of channels and in case of event channels length of the event messages and therefore cannot be determined before execution.

It is strongly recommended to evaluate and test memory consumption for data requests as well as employing additional limitations for end users in the client application that is consuming *ibaHD-API* (e.g. limiting the maximum time range, or the number of samples or signals that can be requested).

Disabling or increasing limits beyond the server hardware capabilities can lead to memory exhaustion of the system and could potentially result in lost data if there are active writers on the *ibaHD-Server* instance.

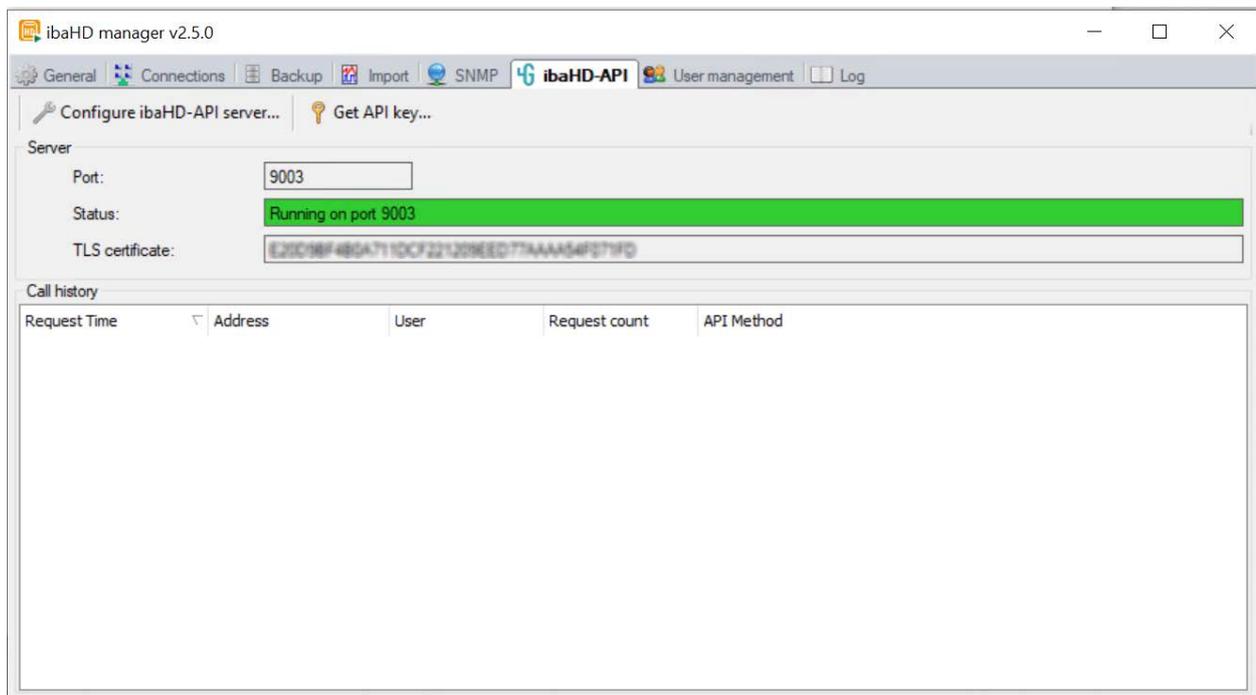
3 ibaHD-API in ibaHD manager

3.1 General information

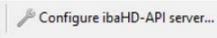
The current *ibaHD-API* configuration is displayed at the tab *ibaHD-API*  at the *ibaHD manager* GUI.

The following information is visualized:

Port	Port number for the <i>ibaHD-API</i> communication
Status	Current status of the API
TLS certificate	Fingerprint of the TLS certificate which is currently in use
Active user API key	API key of current logged in user
Connected clients	List of connected gRPC clients since start of the <i>ibaHD-Server</i> service



3.2 Configuration

By clicking the *Configure ibaHD-API* button  at the top of the tab page the configuration dialog for the *ibaHD-API* settings is displayed.

The dialog is split into three configuration areas.

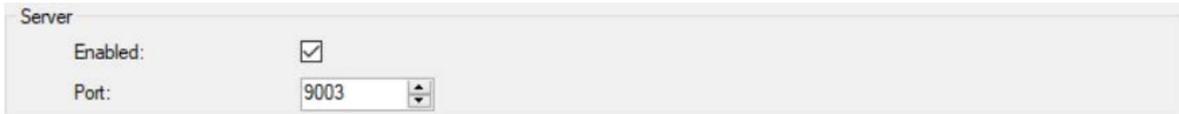
3.2.1 Server

Enable

Enable or disable the *ibaHD-API* functionally

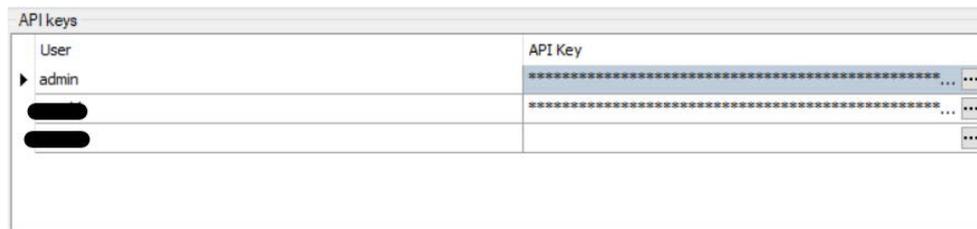
Port

Enter another port if the default port 9003 is in use or another port should be used for the *ibaHD-API* communication

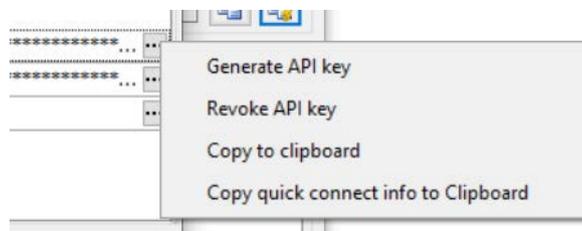


3.2.2 API keys

A list of all the users configured in the ibaHD user management is shown. For each user an API key can be generated in the *API Key* column. This API key is only necessary when user management is enabled.



The button  provides a context menu to manage the user's API key and connection information:



Generate API key

Generate an API key by clicking on the context menu item if no key was created before or if the existing key shouldn't be used anymore. The previous API key will be replaced by the newly generated one.

Revoke API key

Clear the user's API key.

Copy to clipboard

Click this context menu item to copy the user's API key to clipboard. Use this function to transfer the user's API key to your client application.

Copy quick connect info to clipboard

Click this context menu item to copy all user related connect information to the clipboard. The information such as host, certificate API key and user can now be transferred to the client application as text. It can be used to transfer the connection info to ibaDaVIS v2.8.0 or higher.

3.2.3 Certificates

For a safe way of communication, *ibaHD-API* uses X.509 certificates. A certificate is required to communicate through TLS. TLS certificates can be provided by the server and have to be transferred to the client. A *gRPC-API* communication can only take place in case each communication partner trusts the certificate. You can also register certificates and mark them as “not trusted”. The communication with a partner having a “not trusted” certificate will be rejected.

The *Certificates* table displays a list of available certificates. You can manage the certificates here.

You find several buttons in the tool bar of the table. These buttons have the following functions.

Button	Function
	With this button you open a dialog that allows you to load an existing certificate file. The existing certificate has to contain a private key.
	With this button you open a dialog that allows you to generate a new certificate.
	With this button you can export a certificate to a file. You can then register this in an API-client.
	With this button you delete the marked certificate from the table.
	With this button you mark the selected certificate as “trusted”.
	With this button you mark the selected certificate as “not trusted”. However, the certificate remains in the table.
	With this button you select the certificate that should be used for the API communication.

The symbols in the *Properties* column have the following meaning:

Symbol	Meaning
	The certificate is classified as “trusted”.
	The certificate is not trustworthy.
	This certificate is used for the API communication

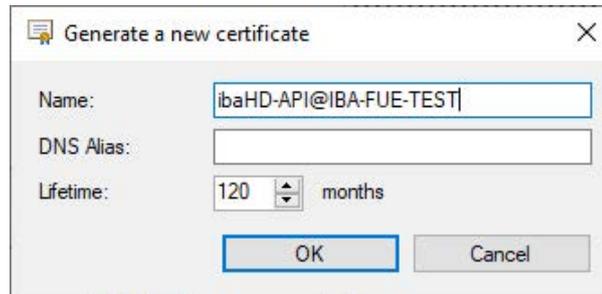
Example



Name	Properties	Issued By	Expiration Date
IbaHD API@localhost	  	IbaHD API@localhost	28.09.2030 11:13:51

3.2.4 Generate a new certificate

1. Click on the button  and the following dialog will open



2. Enter a name of your choice for the certificate or keep the default.
3. Optionally enter a DNS Alias.

By default, the generated certificate will accept connections on all IP addresses that were available on the machine at the time of certificate generation.

To make the API available on another DNS Alias, it can be entered here. The entered alias will then be added to the "Subject alternative name" list of the certificate.

4. Please, define the desired period for the validity (lifetime) of the certificate.
5. Close the dialog with <OK>.

Now, you are asked, if you want to use the certificate you have just generated as API communication certificate.

3.3 Get API key

Click the *Get API key* button  at the top of the tab page to open the following dialog.



In this dialog, you see the API key associated with the currently logged in user (in the example it is the user *admin*).

By clicking the copy button  it is possible to copy the current API key as text to the clipboard. By clicking the quick connect button  the quick connect information is copied to the clipboard.

4 ibaHD-API Functionality

Full details of the API can be found in the *ibaHD-API.proto* file. The proto file is plain text and can be opened with any text editor. It contains the complete structure of the API as well as additional comments about message formats.

The location of the *ibaHD-API.proto* file is in the *ibaHD-Server* installation path in the directory *ibaHD-API*.

The following should give an overview over the APIs capabilities.

If the authentication is not correct, the server will reject the request and answer with an error message.

4.1 GetHdStores()

Allows the client to retrieve a list of HD stores of the server. Time-based and event-based HD stores are supported, length-based HD stores are not included.

Involved Messages

GetHdStoreRequest

Empty structure

GetHdStoreResponse

List of all available time- and event HD stores, including the information about type, enable, active used

4.2 GetHdStoreSchema()

Returns the channel hierarchy for the supplied HD store. It is possible to sort the hierarchy by modules or by logical groups. Event store hierarchies are always returned in a logical group structure (folders).

Involved Messages

GetHdStoreSchemaRequest

The store name, like it is part of the response list from the *GetHdStores* response

GetHdStoreSchemaResponse

List of all available analog and digital channels in a time-based HD store, or all available event names in an event-based HD store; Available additional information in the response can be seen in the proto file

4.3 GetRawChannelData()

This API call provides access to raw and unmodified channel data. Possible use cases are machine learning, providing data to other systems or calculations that require unaggregated data.

The request consists of the time range and the channels for which the data should be retrieved. Additionally, the client can specify the number of samples per message, which otherwise would be auto-determined by the server.

The server always responds with a stream of messages. The *max_sample_count_per_message* field can be used to specify how many samples should be sent in a single message before a new message is started (chunking). This allows the client to control the impact on memory consumption when dealing with large response sizes caused by querying data with high sampling rates.

The time stamp of the first response message does not have to match exactly the start time from the query. This is because the response telegrams return the time stamps that are stored in the *ibaHD-Server* for the requested signal values.

When multiple channels are requested, each response message only contains data for a single channel. The order of data for a single channel is guaranteed. The order of messages for channels is not guaranteed. The server tries to balance the messages equally for all requested channels. It may not be fully possible if for example a channel with low and a channel with high sample rate is requested together.

The following graphic shows the response stream when multiple channels are requested.



The response completes once all messages in the stream have been sent (<https://grpc.io/docs/guides/concepts/#server-streaming-rpc>).

Involved Messages

GetRawChannelDataRequest

Start and end time of the requested time range in Unix timestamp format in microseconds

Channel ids in the same format like they were returned in the response list of *GetHdStoreSchema*

Number of samples per message.

GetRawChannelDataResponse

List of all stored raw channel data in the defined time range with exact time range, split into the different channel formats

Further information can be seen in the proto file

4.4 GetAggregatedChannelData()

GetAggregatedChannelData() provides a way to query aggregated channel data for time-based channels. It allows specifying the number of samples that should be sent for a given time range, as well as the selection of min, max and average aggregations.

The API makes use of *ibaHD-Server* architecture, where each channel is stored in multiple aggregation levels to allow fast and efficient retrieval of data for longer periods of time. Possible use cases are client visualizations (e.g. a monthly average) where the screen size limits the amount of data that can be displayed at once, as well as providing pre-aggregated data for subsequent aggregations where min/max/average data as source is applicable.

Involved Messages

GetAggregatedChannelDataRequest

Start and end time of the requested time range in Unix timestamp format in microseconds

Channel id in the same format like it was returned in the response list of *GetHdStoreSchema*

Number of samples which will be returned

Option to select the aggregated values which will be sent

Further information about possible options can be seen in the proto file

GetAggregatedChannelDataResponse

List of time stamps for the aggregated samples split into the different channel formats

4.5 GetEventData()

Provides all event occurrences for the supplied event channels and the given time range. Additional event fields that are stored together with the event can be optionally selected in the request.

Involved Messages

GetEventDataRequest

Start time and end time of the requested time range in Unix timestamp format in microseconds

Channel ids in the same format like they were returned in the response list of *GetHdStoreSchema*

Further information about possible options can be seen in the proto file

GetEventDataResponse

List of all event occurrences in the defined time range with exact time stamp, message text and so on. Further included information depends on request message and can be seen in the proto file

5 Sample Clients

iba AG provides sample clients for the following programming languages:

- C#
- C++
- Python

All sample clients are designed to demonstrate a minimal setup for accessing ibaHD-API, they do not cover all API functionality.

The sample clients are available on GitHub:

<https://github.com/iba-ag/ibaHD-API-Sample-Clients>

Disclaimer

This code is provided “as is” and serves as an example for the usage of ibaHD-API. The integration of ibaHD-API in external systems and general support for programming languages is not provided by iba AG. All terms of ibaHD-Server's license agreement apply.

6 Support and contact

Support

Phone: +49 911 97282-14
Fax: +49 911 97282-33
Email: support@iba-ag.com

Note



If you require support, indicate the serial number (iba-S/N) of the product or the license number.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0
Fax: +49 911 97282-33
Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

www.iba-ag.com.