

ibaHD-Server

Digitaler iba-Tag | USERGROUP





Gerald Schöppner
Produktmanager



Frederic Lesemeister
Applikation & Consulting



Fragen / Diskussionsbeiträge

Bitte Mikrofon freigeben und direkt im Plenum Ihre Frage stellen.



Oder nutzen Sie die Chat Funktion. Ihre Frage wird dann im Anschluss an den Vortrag beantwortet.



Kamera

Wir freuen uns über jedes Gesicht, welches wir sehen.



Aufzeichnung

Die Session wird nicht aufgezeichnet.



Präsentation

Im Anschluss an den digitalen iba-Tag stehen die Präsentationen der Usergroups und der Vortrag „Produktneuheiten“ auf <https://www.iba-ag.com/de/iba-tag> zur Verfügung.

Sie werden darüber per E-Mail informiert.

ibaTag 2021 Session ibaHD-Server

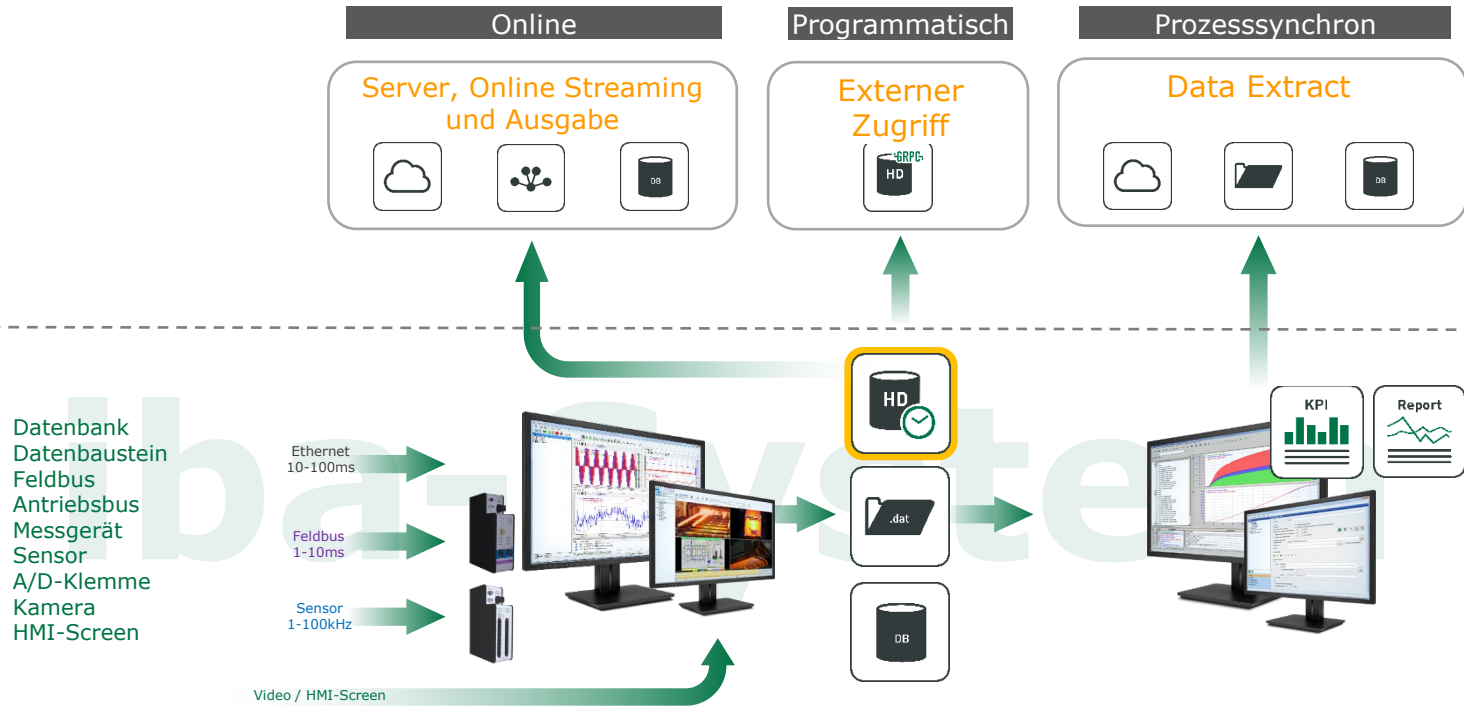
Das erwartet Sie in dieser Session:

- **Datenablage im ibaHD-Server**
 - Kontinuierliche Datenaufzeichnung
 - Events (Online / Offline)
- HD-Daten in der iba Systemlandschaft
- ibaHD-API über BloomRPC
 - Einrichtung / Konfiguration
 - Abfrage von HD-Daten über BloomRPC
- Verarbeitung von HD-Daten mittels Python
- Zeit für Ihre Fragen

[Live DEMO](#)[Live DEMO](#)[Live DEMO](#)

Datenaufzeichnung im ibaHD-Server

Einordnung Systemtopologie



Datenaufzeichnung im ibaHD-Server

kontinuierliche Aufzeichnung / Events



Getriggert:

Zeitzyklisch
Auftragsbezogen
Produktbezogen



Kontinuierlich

Events (Beispiele)

Aufträge
Produkte
Störungen
....



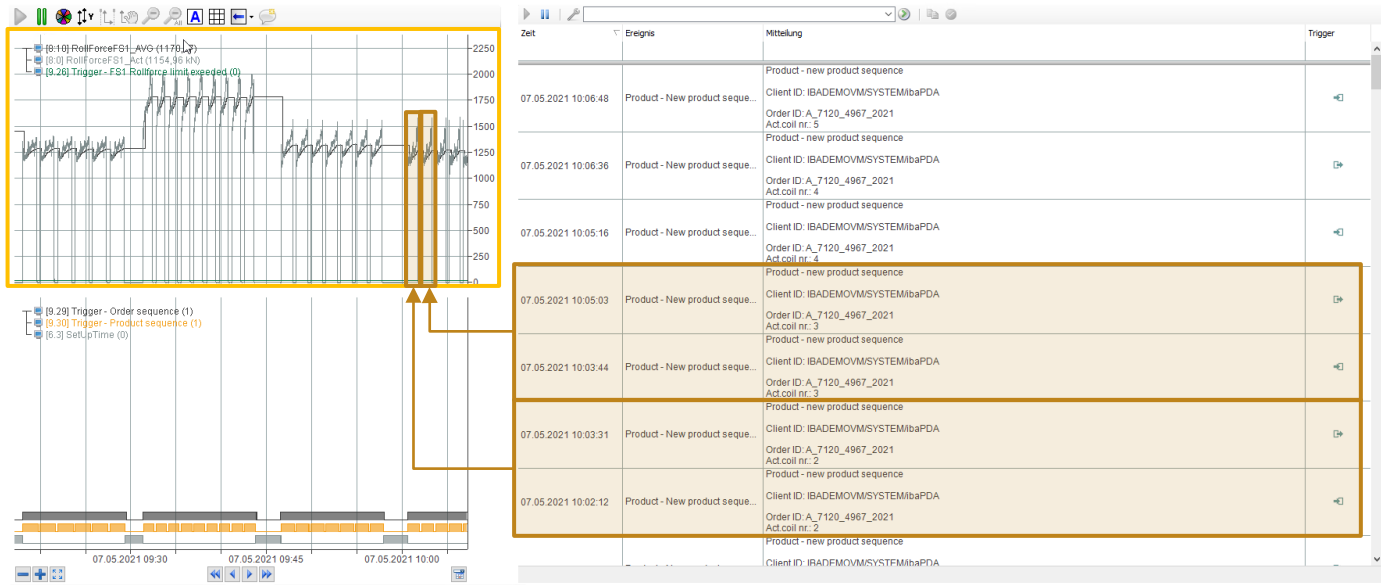
Datenaufzeichnung im ibaHD-Server

Visualisierung der ibaHD-Daten (ibaPDA)



Produktion > Produkte / Aufträge

Coil ausschleusen



Datenaufzeichnung im ibaHD-Server

Konfiguration der Datenaufzeichnung



ibaHD-Datenaufzeichnung 1 (QPanel) - Puffer

Speicherpuffer

Maximale Größe: MB Leere Konfiguration anlegen:

Speicherpuffer regelmäßig sichern alle s

Dateipuffer

Dateipuffer verwenden Aktuelle Dateikonfiguration:

Dateiverzeichnis:

Maximale Größe: GB

Weitere Puffereinstellungen

Maximale Zeit: Stunden

Diagnose Speicherpuffer	Diagnose Dateipuffer
Zuletzt entnommes Element: <input type="text" value="07.05.2021 11:07:19.709"/>	Zuletzt entnommes Element: <input type="text"/>
Füllstand: <input type="text" value="0,0 %"/>	Füllstand: <input type="text" value="0,0 %"/>
	Unverarbeiteter Anteil: <input type="text" value="0,0 %"/>

Konfigurationsmöglichkeiten bei der HD-Datenaufzeichnung:

- Signale / Aufzeichnungszyklen
- Online Ereignisse
 - Unterschiedliche Ereignisse (Auftrag / Produkt / Störung / etc.)
 - Mitteilungen je Ereignis (analog / digital / String)
- Pufferungsmöglichkeiten

Live Demo ibaPDA

ibaTag 2021 Session ibaHD-Server

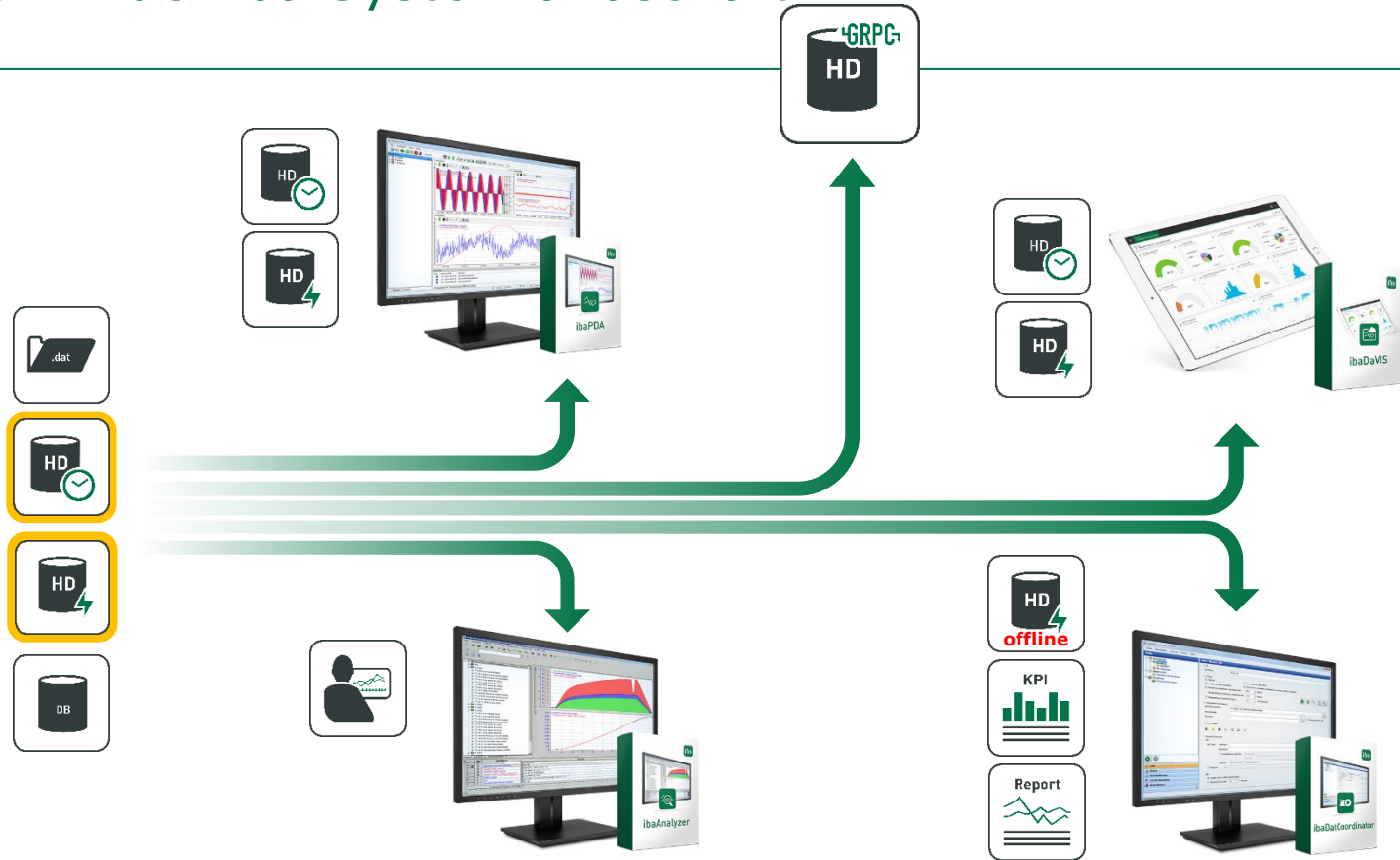
Das erwartet Sie in dieser Session:

- Datenablage im ibaHD-Server
 - Kontinuierliche Datenaufzeichnung
 - Events (Online / Offline)
- **HD-Daten in der iba Systemlandschaft**
- ibaHD-API über BloomRPC
 - Einrichtung / Konfiguration
 - Abfrage von HD-Daten über BloomRPC
- Verarbeitung von HD-Daten mittels Python
- Zeit für Ihre Fragen

[Live DEMO](#)[Live DEMO](#)[Live DEMO](#)

HD-Daten in der iba Systemlandschaft

Beispiele



Live Demo ibaDatCoordinator

ibaTag 2021 Session ibaHD-Server

Das erwartet Sie in dieser Session:

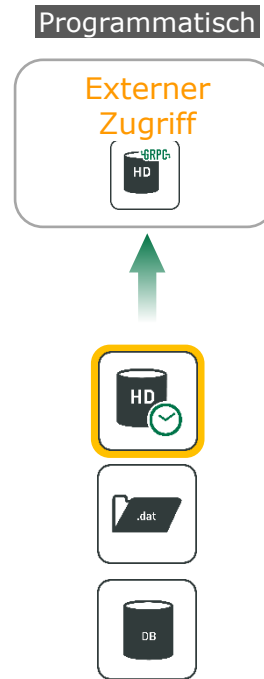
- Datenablage im ibaHD-Server
 - Kontinuierliche Datenaufzeichnung
 - Events (Online / Offline)
- HD-Daten in der iba Systemlandschaft
- **ibaHD-API über BloomRPC**
 - Einrichtung / Konfiguration
 - Abfrage von HD-Daten über BloomRPC
- Verarbeitung von HD-Daten mittels Python
- Zeit für Ihre Fragen

[Live DEMO](#)[Live DEMO](#)[Live DEMO](#)

Das iba-System – Ein offenes Ecosystem



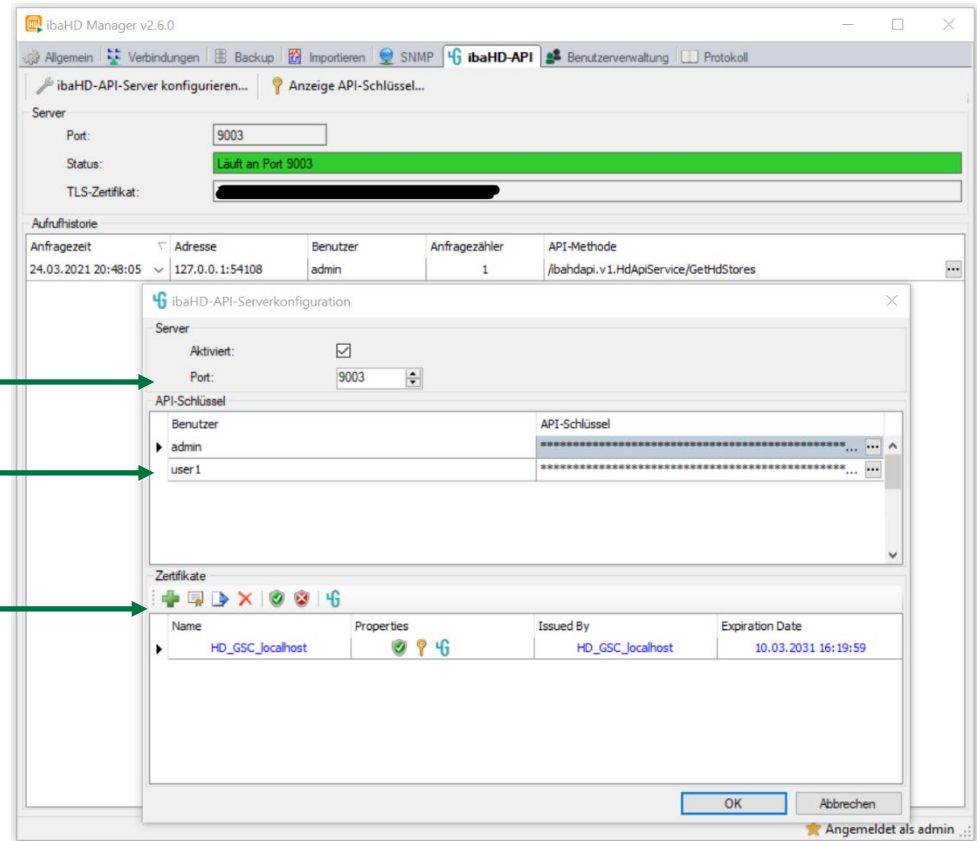
- Gesicherter Verbindungsaufbau über Zertifikatsdatei und Access-Key
 - Zugriffsbeschränkung über vorhandenes User-Management
 - Verschlüsselte Datenübertragung
 - Request – Response Konzept
-
- Abfrage der vorhandenen Stores (Data- und Eventstores)
 - Abfrage der Signal- / Event-Namen
 - Abfrage von Signalwerten über einen definierten Zeitbereich als aggregierte Werte oder Rohwerte
 - Abfrage von Ereignis innerhalb eines definieren Zeitbereiches



API = **a**pplication **p**rogramming **i**nterface
(Anwendungsprogrammierschnittstelle)

gRPC = **g**RPC **R**emote **P**rocedure **C**alls

- Konfiguration der Portnummer
- Vergabe der API Schlüssel
- Erstellung und Verwaltung von Zertifikatsdateien



ibaHD-Server

- 1x Daten Ablage mit Messdaten aus einem Playbackfile
- 1x Event Ablage

BloomRPC Demo

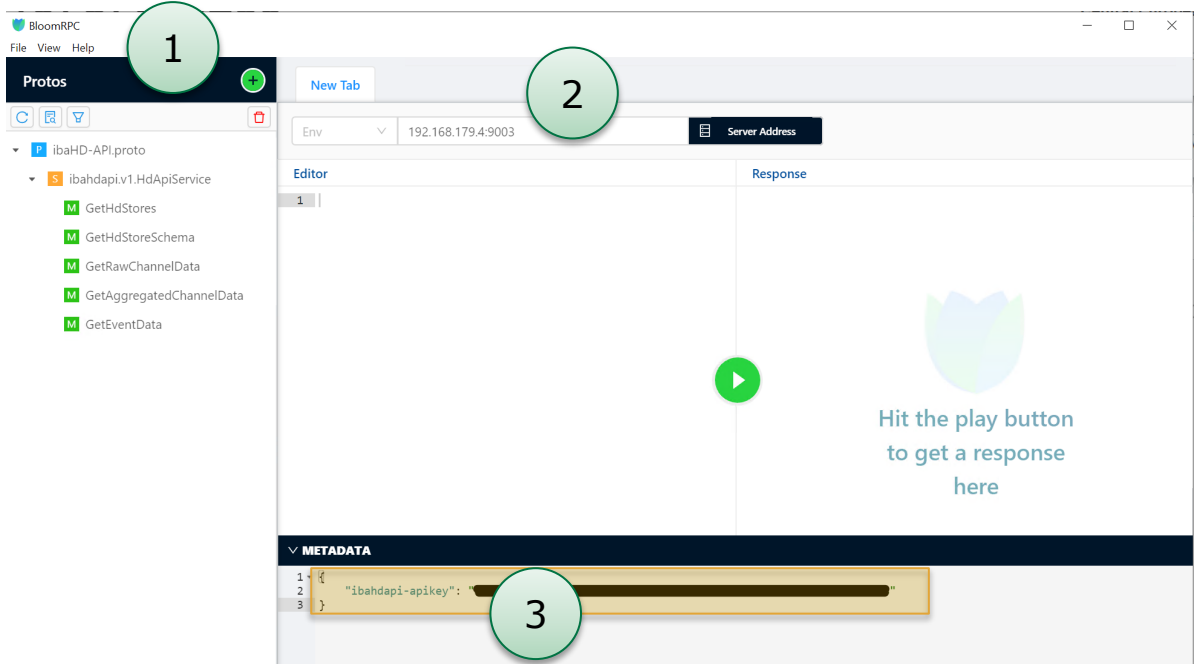
- BloomRPC einrichten
- Abfrage GetHdStores
- Abfrage GetHDStoreSchema
- Abfrage GetRawChannelData
- Abfrage GetEventData

Python Demo

- Client vorbereiten
- Raw Daten Abfrage
- Export nach Excel
- Event Abfrage

BloomRPC einrichten

1. ‚proto‘ Datei laden
2. Server Adresse eingeben
3. Metadata mit API-Key eintragen



Abfrage GetHdStore

1. GetHdStores auswählen
2. TLS auswählen
3. Zertifikat eintragen bzw. auswählen
4. Play drücken

The screenshot shows the BloomRPC interface. On the left, the Protos list is expanded to show the `GetHdStores` method under `ibahdapi.v1.HdApiService`, highlighted with a green circle labeled '1'. The main editor shows the unary call `ibahdapi.v1.HdApiService.GetHdStores` with the URL `192.168.179.4:9003`. The response is displayed in a JSON format, with a green circle labeled '2' over the response area. A green play button labeled '4' is positioned over the response area. A dialog box titled 'TLS / SSL Manager' is open in the foreground, showing the 'Add Root Certificate' section with a radio button selected for 'NOTE565.crt', and a green circle labeled '3' over the 'Import Key' and 'Import Cert Chain' options.

```
Editor  
1 {}  
  
Response  
0.040s  
{  
  "hd_stores": [  
    {  
      "hd_store_name": "ibaTag2021-Data",  
      "hd_store_type": "HD_STORE_TYPE_TIME",  
      "enabled": true,  
      "active": true  
    },  
    {  
      "hd_store_name": "ibaTag2021-Event",  
      "hd_store_type": "HD_STORE_TYPE_EVENT",  
      "enabled": true,  
      "active": true  
    }  
  ]  
}
```

TLS / SSL Manager
Add Root Certificate
Root Certificate Private Key Cert Chain
Server Certificate - -
NOTE565.crt Import Key Import Cert Chain ssl target host X
Close Done

Abfrage GetHdStoreSchema

1. GetHdStoreSchema auswählen
2. TLS auswählen
3. Zertifikat eintragen bzw. auswählen
4. Name der gewünschten Ablage eintragen (z.B. Kopie von GetHdStores)
5. Play drücken

The screenshot shows the BloomRPC interface with the following elements:

- Protos List (Step 1):** A list of protocols under 'ibahd-API.proto' and 'ibahdapi.v1.HdApiService'. 'GetHdStoreSchema' is selected, indicated by a green circle '1'.
- TLS Selection (Step 2):** The 'Unary Call' tab is active, and the 'TLS' checkbox is checked, indicated by a green circle '2'.
- Request Editor (Step 4):** The request body is a JSON object:

```
{ "hd_store_name": "ibaTag2021-Data", "sort_by": 0, "info_fields": true }
```

. The value 'ibaTag2021-Data' is highlighted with a yellow box and a green circle '4'.
- Response Viewer (Step 5):** The response is a large JSON object. A green play button (Step 5) is positioned to the left of the response area.
- TLS/SSL Manager (Step 3):** A dialog box titled 'TLS / SSL Manager' is open, showing the 'Add Root Certificate' section. The 'NOTE565.crt' certificate is selected, indicated by a green circle '3'.

Abfrage GetRawChannelData

1. GetRawChannelData auswählen
2. TLS auswählen
3. Zertifikat eintragen bzw. auswählen
4. Start und Ende Zeit der Abfrage eintragen (in Mikrosekunden)
5. Name der gewünschten Signale eintragen (z.B. Kopie von GetHdStoreSchema)
6. Play drücken

The screenshot shows the BloomRPC interface with several components:

- Protos sidebar (1):** A tree view showing the protocol definition. The path is `ibahd.v1.HdApiService` > `GetRawChannelData`.
- Editor (4):** A text editor showing the query configuration. The JSON body is:

```
1- {
2  "time_range_from": 1619788230000000,
3  "time_range_to": 1619788240000000,
4  "channel_id": "ibaTag2021-Data\\[0:0]",
5  "ibaTag2021-Data\\[0:0]",
6  "ibaTag2021-Data\\[0:1]"
7 },
8 "max_sample_size": 20
9 }
```
- TLS/SSL Manager (3):** A dialog box for configuring TLS. The `NOTE565.crt` certificate is selected under the `Import Key` and `Import Cert Chain` options.
- Stream viewer (6):** A panel showing the response data for Stream 1. The response is a JSON object with `float_values`, `double_values`, `channel_id`, `start_timestamp`, `step`, `data_type`, `string_values`, and `digital_values`.

Abfrage GetAggregatedChannelData

1. GetAggregatedChannelData auswählen
2. TLS auswählen
3. Zertifikat eintragen bzw. auswählen
4. Start und Ende Zeit der Abfrage eintragen (in Mikrosekunden)
5. Name der gewünschten Signale eintragen (z.B. Kopie von GetHdStoreSchema)
6. Anzahl der Aggregations-ergebnisse eintragen
7. Play drücken

The screenshot shows the BloomRPC interface with the following components:

- Protos Panel:** Shows the hierarchy of protocols, with `GetAggregatedChannelData` selected under `ibahdapi.v1.HdApiService`.
- TLS/SSL Manager:** A dialog box for adding a root certificate, with `NOTE565.crt` selected.
- Editor:** The main configuration area for the query, showing the following JSON structure:

```
1 - {
2   "time_range_from": 1619788230000000,
3   "time_range_to": 1619799030000000,
4   "channel_ids": [
5     "ibaTag2021-Data\\[2:18]",
6     "ibaTag2021-Data\\[2:19]"
7   ],
8   "sample_count": 18,
9   "min_aggregation": true,
10  "max_aggregation": true,
11  "avg_aggregation": true
12 }
```
- Response:** A preview of the query results, showing a list of timestamps and aggregated channel data for the specified channels.

Abfrage GetEventData

1. GetEventData auswählen
2. TLS auswählen
3. Zertifikat eintragen bzw. auswählen
4. Start und Ende Zeit der Abfrage eintragen (in Mikrosekunden)
5. Name der gewünschten Events eintragen (z.B. Kopie von GetHdStoreSchema)
6. Namen der gewünschten Zusatzfelder eingeben
7. Play drücken

The screenshot shows the BloomRPC interface with the following elements:

- Protos sidebar:** A tree view showing the protocol structure. Step 1 points to the `GetEventData` method.
- Unary Call:** A button to execute the query. Step 2 points to the TLS icon.
- Editor:** A JSON query editor. Step 4 points to the `time_range_from` and `time_range_to` fields. Step 5 points to the `channel_ids` field. Step 6 points to the `numeric_fields` field.
- Response:** A pane showing the JSON response. Step 7 points to the play button.
- TLS / SSL Manager:** A dialog box for adding certificates. Step 3 points to the `NOTE565.crt` certificate.

ibaTag 2021 Session ibaHD-Server

Das erwartet Sie in dieser Session:

- Datenablage im ibaHD-Server
 - Kontinuierliche Datenaufzeichnung
 - Events (Online / Offline)
- Verarbeitung der ibaHD-Daten im ibaDatCoordinator
- Verwendung der ibaHD-API über BloomRPC
 - Einrichtung / Konfiguration
 - Abfrage von HD-Daten
- **Verarbeitung von HD-Daten mittels Python**
- Zeit für Ihre Fragen



Beispiel Client vorbereiten

1. ibahdapi_sampleclient.py herunterladen
2. Anweisung in Readme.md ausführen
3. Pfad zur Zertifikationsdatei eintragen
4. API-Key eintragen
5. Server Adresse eingeben

1

3

4

5

```
ibahdapi_sampleclient.py X
C:\> dat > ibaTag2021 > Python > ibahdapi_sampleclient.py > run
1 from __future__ import print_function
2 import logging
3 import grpc
4 import ibahd_API_pb2
5 import ibahd_API_pb2_grpc
6
7 # Helper class to apply the api key to each call
8 class ApiKeyCallCredentials(grpc.AuthMetadataPlugin):
9
10     def __init__(self, apikey):
11
12         self._apikey = apikey
13
14     def __call__(self, context, callback):
15
16         metadata = (('ibahdapi-apikey', self._apikey),)
17
18         callback(metadata, None)
19
20 def run():
21
22     # Read exported (PEM formatted) server TLS certificate from disk
23     certificate = open('C:\\dat\\ibaTag2021\\Cert\\NOTE565.crt', 'rb').read()
24     tls_credentials = grpc.ssl_channel_credentials(certificate)
25
26     # Apply the api key to every request that is made with the client
27     apikey_credentials = grpc.metadata_call_credentials(ApiKeyCallCredentials(''))
28
29     combined_credentials = grpc.composite_channel_credentials(tls_credentials, apikey_credentials)
30
31     # gRPC channel setup to connect to ibahd-API endpoint in ibahd-Server
32     # Increasing the default message size (~4MB) is recommended (c int32_max = 2147483647)
33     with grpc.secure_channel('127.0.0.1:9003', combined_credentials, options=[('grpc.max_receive_message_length', 2147483647)]) as channel:
```

Abfrage GetHdStore

1. Code aus Beispielclient starten
2. Ausgabe auf Terminal

```
# gRPC channel setup to connect to ibaHD-API endpoint in ibaHD-Server
# Increasing the default message size (~4MB) is recommended (c int32 max = 2147483647)
with grpc.secure_channel('127.0.0.1:9003', combined_credentials, options=[('grpc.max_receive_message_length', 2147483647)]) as channel:

    # Instantiate ibaHD-API client on the gRPC channel
    client = ibaHD_API_pb2_grpc.HdApiServiceStub(channel)

    # Simple request to retrieve stores
    response = client.GetHdStores(request=ibaHD_API_pb2.GetHdStoresRequest())

    # Clean channel shutdown
    channel.close()

print(response)
```

1

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\dat\ibaTag2021\Python> &

hd_stores {
  hd_store_name: "ibaTag2021-Data"
  hd_store_type: HD_STORE_TYPE_TIME
  enabled: true
  active: true
}
hd_stores {
  hd_store_name: "ibaTag2021-Event"
  hd_store_type: HD_STORE_TYPE_EVENT
  enabled: true
  active: true
}

PS C:\dat\ibaTag2021\Python>
```

2

Abfrage Raw Daten

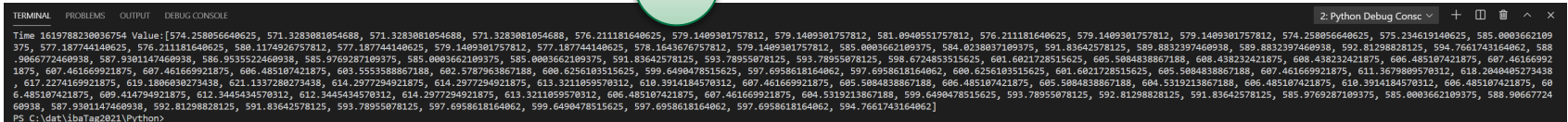
1. Python Funktion zur Eingabe der Start und Ende Zeit
2. Python ,setattr' Funktion nutzen um Parameter für Ausgabe einzutragen
3. Ausgabe auf Terminal

```
47 start_time_epoch = int(datetime.datetime(2021,4,30,15,10,30,0).timestamp() * 1e6)
48 end_time_epoch = int(datetime.datetime(2021,4,30,15,10,40,0).timestamp() * 1e6)
49
50
51 channelname = ["ibaTag2021-Data\\[2:18]"]
52 xparam = ibaHD_API_pb2.GetRawChannelDataRequest(channel_ids = channelname)
53 setattr(xparam,"time_range_from",start_time_epoch)
54 setattr(xparam,"time_range_to",end_time_epoch)
55 setattr(xparam,"max_sample_count_per_message",100)
56
57 responsevalue = client.GetRawChannelData(request=xparam)
58 for data in responsevalue:
59     print (f"Time {data.start_timestamp} Value:{data.float_values}")
60
```

1

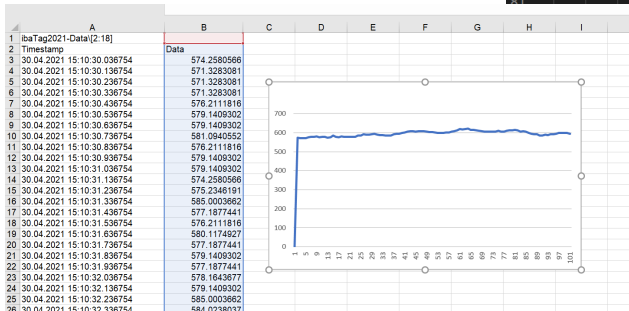
2

3



Raw Daten nach Excel schreiben

1. Daten der Antwort in Excel Zellen schreiben



```

57 responsevalue = client.GetRawChannelData(request=xparam)
58
59 if (end_time_epoch-start_time_epoch) <= 10000000:
60     wb = xlwt.Workbook()
61     ws = wb.add_sheet('Table1')
62     ws.write(1,0,"Timestamp")
63     ws.write(1,1,"Data")
64     lastline = 2
65     channel_id = ""
66
67     for data in responsevalue:
68
69         # write data into excel sheet
70         output_timestamp = data.start_timestamp
71         output_offset = data.step
72         channel_id = data.channel_id
73
74         for newvalues in data.float_values:
75             if lastline >= 65500:
76                 break
77             valuetimestamp = datetime.datetime.fromtimestamp(output_timestamp / 1e6)
78             ws.write(lastline, 0, str(valuetimestamp.day).zfill(2)+"."+str(valuetimestamp.month).zfill(2)+"." \
79                 + str(valuetimestamp.year).zfill(4)+" " + str(valuetimestamp.hour).zfill(2)+":" \
80                 + str(valuetimestamp.minute).zfill(2)+":" + str(valuetimestamp.second).zfill(2)+"." \
81                 + str(valuetimestamp.microsecond).zfill(6))
82             ws.write(lastline, 1, newvalues)
83             output_timestamp += output_offset
84             lastline += 1
85
86     ws.write(0,0, channel_id)
87     wb.save('SampleValues.xls')

```



Abfrage GetEventData

1. Python Funktion zur Eingabe der Start und Ende Zeit
2. Python ‚setattr‘ Funktion nutzen um Parameter für Ausgabe einzutragen
3. Ausgabe auf Terminal

```
93 # read events for 30 min and print them on the terminal
94
95 start_time_epoch = int(datetime.datetime(2021,4,30,15,10,30,0).timestamp() * 1e6)
96 end_time_epoch = int(datetime.datetime(2021,4,30,15,40,30,0).timestamp() * 1e6)
97
98 channelname = ["ibaTag2021-Event\\[1:2]"]
99 eventparam = ibaHD_API_pb2.GetEventDataRequest(channel_ids = channelname)
100 setattr(eventparam,"time_range_from",start_time_epoch)
101 setattr(eventparam,"time_range_to",end_time_epoch)
102 setattr(eventparam,"limit_per_channel",100)
103
104 response_event = client.GetEventData(request=eventparam)
105
106 print (response_event)
107
```

1

2

3

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
events {
  channel_id: "ibaTag2021-Event\\[1:2]"
  timestamp: 1619790028537371
  message: "F1 Ger\303\274st belegt AS: 24.44 || BS: 35.19\nProduktID: 037180 "
  trigger: EVENT_TRIGGER_TYPE_OUTGOING
}
events {
  channel_id: "ibaTag2021-Event\\[1:2]"
  timestamp: 1619789934837456
  message: "F1 Ger\303\274st belegt AS: 696.33 || BS: 893.60\nProduktID: 037180 "
  trigger: EVENT_TRIGGER_TYPE_INCOMING
}
events {
  channel_id: "ibaTag2021-Event\\[1:2]"
  timestamp: 1619789921837427
  message: "F1 Ger\303\274st belegt AS: 25.42 || BS: 30.30\nProduktID: 024760 "
  trigger: EVENT_TRIGGER_TYPE_OUTGOING
}
events {
  channel_id: "ibaTag2021-Event\\[1:2]"
  timestamp: 1619789851537366
  message: "F1 Ger\303\274st belegt AS: 619.18 || BS: 775.43\nProduktID: 024760 "
  trigger: EVENT_TRIGGER_TYPE_INCOMING
}
```

ibaTag 2021 Session

ibaHD-Server

Das erwartet Sie in dieser Session:

- Datenablage im ibaHD-Server
 - Kontinuierliche Datenaufzeichnung
 - Events (Online / Offline)
- Verarbeitung der ibaHD-Daten im ibaDatCoordinator
- Verwendung der ibaHD-API über BloomRPC
 - Einrichtung / Konfiguration
 - Abfrage von HD-Daten
- Verarbeitung von HD-Daten mittels Python
- **Zeit für Ihre Fragen**

