



ibaPDA-Interface-HTTP(S)

Datenschnittstelle für HTTP(S)

Handbuch
Ausgabe 1.0

Messsysteme für Industrie und Energie
www.iba-ag.com

Hersteller

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland

Kontakte

| | |
|----------|--|
| Zentrale | +49 911 97282-0 |
| Support | +49 911 97282-14 |
| Technik | +49 911 97282-13 |
| E-Mail | iba@iba-ag.com |
| Web | www.iba-ag.com |

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zu widerhandlungen verpflichten zu Schadenersatz.

© iba AG 2024, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website www.iba-ag.com zum Download bereit.

| Version | Datum | Revision | Autor | Version SW |
|---------|---------|-------------|-------|------------|
| 1.0 | 07-2024 | Erstausgabe | st | 8.7.0 |

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

Inhalt

| | | |
|----------|--|-----------|
| 1 | Zu dieser Dokumentation | 4 |
| 1.1 | Zielgruppe und Vorkenntnisse | 4 |
| 1.2 | Schreibweisen | 4 |
| 1.3 | Verwendete Symbole | 5 |
| 2 | Systemvoraussetzungen | 6 |
| 3 | Die HTTP(S)-Schnittstelle | 7 |
| 3.1 | Systemtopologien | 7 |
| 3.2 | Konfiguration und Projektierung ibaPDA | 8 |
| 3.2.1 | Einstellungen der Schnittstelle | 8 |
| 3.2.2 | Modul hinzufügen | 9 |
| 3.2.3 | Allgemeine Moduleinstellungen | 10 |
| 3.2.4 | Request-Einstellungen | 12 |
| 3.2.5 | Register Response | 20 |
| 3.2.6 | Analoge und digitale Signale | 24 |
| 4 | Anwendungsbeispiel HTTP(S) | 25 |
| 4.1 | Vorbereitung | 25 |
| 4.2 | Konfiguration des ibaPDA HTTP(S)-Moduls | 26 |
| 4.3 | Weiterverarbeitung der abgefragten Daten | 30 |
| 4.4 | Texttrenner-Modul konfigurieren | 30 |
| 5 | Diagnose | 32 |
| 5.1 | Lizenz | 32 |
| 5.2 | Protokolldateien | 32 |
| 5.3 | Fehlermeldungen | 33 |
| 5.4 | Verbindungsdiagnose mittels PING | 34 |
| 5.5 | Diagnosemodule | 35 |
| 6 | Support und Kontakt | 41 |

1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und Anwendung der Software-Schnittstelle *ibaPDA-Interface-HTTP(S)*.

Hinweis



Diese Dokumentation ist eine Ergänzung zum *ibaPDA*-Handbuch. Informationen über alle weiteren Eigenschaften und Funktionen von *ibaPDA* finden Sie im *ibaPDA*-Handbuch bzw. in der Online-Hilfe.

1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation wendet sich an ausgebildete Fachkräfte, die mit dem Umgang mit elektronischen und elektronischen Baugruppen sowie der Kommunikations- und Messtechnik vertraut sind. Als Fachkraft gilt, wer auf Grund seiner fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die ihm übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

Für den Umgang mit *ibaPDA-Interface-HTTP(S)* sind folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Betriebssystem Windows
- Grundkenntnisse *ibaPDA*
- Grundkenntnisse des HTTP(S)-Protokolls

1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

| Aktion | Schreibweise |
|-------------------------------------|---|
| Menübefehle | Menü <i>Funktionsplan</i> |
| Aufruf von Menübefehlen | <i>Schritt 1 – Schritt 2 – Schritt 3 – Schritt x</i> Beispiel: Wählen Sie Menü <i>Funktionsplan – Hinzufügen – Neuer Funktionsblock</i> |
| Tastaturtasten | <Tastenname> Beispiel: <Alt>; <F1> |
| Tastaturtasten gleichzeitig drücken | <Tastenname> + <Tastenname> Beispiel: <Alt> + <Strg> |
| Grafische Tasten (Buttons) | <Tastenname> Beispiel: <OK>; <Abbrechen> |
| Dateinamen, Pfade | <i>Dateiname, Pfad</i> Beispiel: <i>Test.docx</i> |

1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:

Gefahr!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!

- Beachten Sie die angegebenen Maßnahmen.
-

Warnung!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!

- Beachten Sie die angegebenen Maßnahmen.
-

Vorsicht!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!

- Beachten Sie die angegebenen Maßnahmen.
-

Hinweis



Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

Tipp



Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

Andere Dokumentation



Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

2 Systemvoraussetzungen

Folgende Systemvoraussetzungen sind für die Verwendung der Datenschnittstelle HTTP(S) erforderlich:

- *ibaPDA v8.7.0 oder höher*
- Lizenz für *ibaPDA-Interface-HTTP(S)*
- Internet-Verbindung

Sonstige Voraussetzungen an die eingesetzte Rechner-Hardware und die unterstützten Betriebssysteme entnehmen Sie bitte der *ibaPDA*-Dokumentation.

| Bestellnr. | Produktbezeichnung | Beschreibung |
|------------|--------------------------|--|
| 31.001018 | ibaPDA-Interface-HTTP(S) | Erweiterungslizenz für ein ibaPDA-System mit der Datenschnittstelle HTTP(S), keine Begrenzung der Verbindungen |

3 Die HTTP(S)-Schnittstelle

Mit der Datenschnittstelle *ibaPDA-Interface-HTTP(S)* bietet *ibaPDA* die Möglichkeit, Abfragen mittels HTTP(S) zu konfigurieren und Daten über HTTP(S) zu empfangen. Anwender können damit Werte via HTTP(S) von Webservices anfordern und dorthin schreiben. Das Ergebnis kann in einem Textsignal weiterverarbeitet werden.

Diese HTTP-Methoden werden unterstützt: Get, Post, Put, Delete, Patch, Options, Trace, Head.

Abfragen können zyklisch oder ereignisbasiert gesendet werden.

3.1 Systemtopologien

Die Verbindungen zu den Webservern können über Standard-Ethernet-Schnittstellen des Rechners hergestellt werden.

Es ist keine weitere Software für den Betrieb erforderlich.

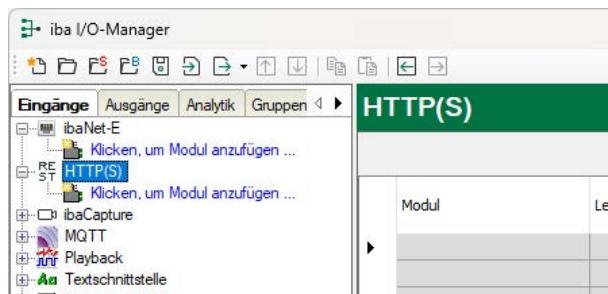
Hinweis



Es wird empfohlen, die TCP/IP-Kommunikation auf einem separaten Netzwerksegment durchzuführen, um eine gegenseitige Beeinflussung durch sonstige Netzwerkkomponenten auszuschließen.

3.2 Konfiguration und Projektierung ibaPDA

Nachfolgend ist die Projektierung in *ibaPDA* beschrieben. Wenn alle Systemvoraussetzungen erfüllt sind, bietet *ibaPDA* im Schnittstellenbaum des I/O-Managers die Schnittstelle *HTTP(S)* an.



Hinweis



Ein Anwendungsbeispiel für eine *HTTP(S)*-Abfrage finden Sie in Kapitel **Anwendungsbeispiel *HTTP(S)***, Seite 25 . Das Beispiel erläutert anhand einer Abfrage, die in java script formuliert ist, die Konfiguration des *HTTP(S)*-Moduls sowie die Weiterverarbeitung und Darstellung der abgefragten Daten.

3.2.1 Einstellungen der Schnittstelle

Die Schnittstellenübersicht enthält eine Liste der konfigurierten *HTTP(S)*-Module mit Statistikinformationen für jedes Modul.

| HTTP(S) | | | | | | | | |
|---------|------------|--------------|------------|-------------|-----|-----|-----------------------|------------------------|
| Modul | Lesezähler | Fehlerzähler | Datengröße | Antwortzeit | | | Protokolldatei öffnen | Statistik zurücksetzen |
| | | | | Actual | Min | Max | | |
| ? | ? | ? | ? | ? | ? | ? | | |
| ? | ? | ? | ? | ? | ? | ? | | |
| ? | ? | ? | ? | ? | ? | ? | | |
| ? | ? | ? | ? | ? | ? | ? | | |

Verbindungstabelle

Die Tabelle zeigt die konfigurierten Verbindungen. Jede Tabellenzeile entspricht einem *HTTP(S)*-Modul bzw. einer Verbindung.

<Protokolldatei öffnen>

Wenn *HTTP(S)*-Verbindungen hergestellt wurden, dann werden alle verbindungsspezifischen Aktionen in einer Textdatei protokolliert. Diese Datei können Sie über diesen Button öffnen und einsehen. Im Dateisystem auf der Festplatte finden Sie die Protokolldatei im Programmmpfad von *ibaPDA*-Server (`C:\ProgramData\iba\ibaPDA\Log`). Der Dateiname der aktuellen Protokolldatei lautet `httpLog.txt`, der Name der archivierten Protokolldateien lautet `httpLog_yyyy_mm_dd_ss.txt`.

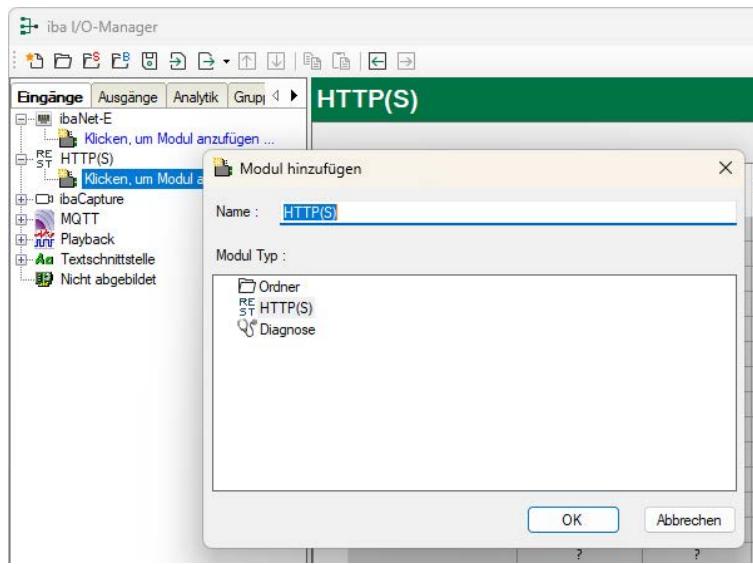
<Statistik zurücksetzen>

Wenn Sie die Zähler für alle Verbindungen zurücksetzen möchten, klicken Sie auf die Schaltfläche <Statistik zurücksetzen>.

3.2.2 Modul hinzufügen

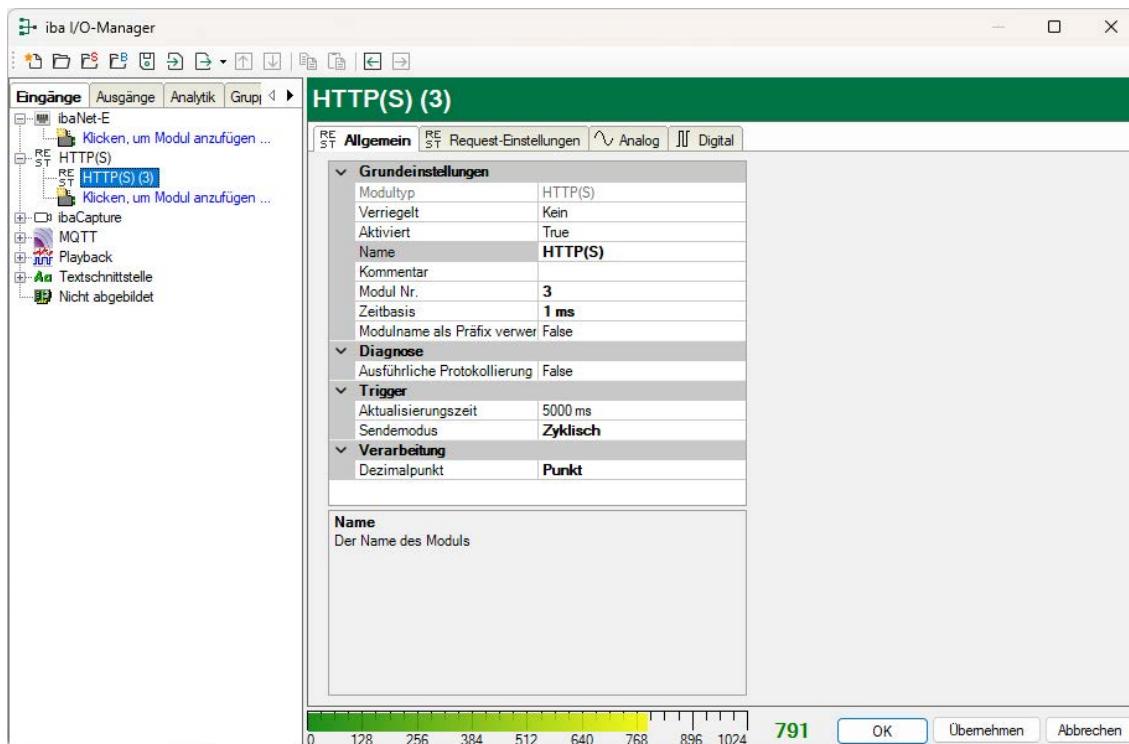
Für jede Verbindung wird ein HTTP(S)-Modul benötigt. Sie können auch ein oder mehrere Diagnosemodule hinzufügen. Weitere Informationen zu den Diagnosemodulen finden Sie in Kapitel **»Diagnosemodule**, Seite 35.

1. Fügen Sie ein Modul hinzu, indem Sie auf den blauen Befehl *Klicken, um Modul anzufügen* unter der Schnittstelle klicken.
2. Wählen Sie den Modultyp HTTP(S) aus, vergeben bei Bedarf einen Namen und klicken auf <OK>.



3.2.3 Allgemeine Moduleinstellungen

Im Register *Allgemein* können folgende Moduleinstellungen konfiguriert werden:



Grundeinstellungen

Modultyp (nur Anzeige)

Zeigt den Typ des aktuellen Moduls an.

Verriegelt

Sie können ein Modul verriegeln, um ein versehentliches oder unautorisiertes Ändern der Einstellungen zu verhindern.

Aktiviert

Aktivieren Sie das Modul, um Signale aufzuzeichnen.

Name

Hier können Sie einen Namen für das Modul eintragen.

Kommentar

Hier können Sie einen Kommentar oder eine Beschreibung zum Modul eintragen. Dies wird dann als Tooltip im Signalbaum angezeigt.

Modul Nr.

Diese interne Referenznummer des Moduls bestimmt die Reihenfolge der Module im Signalbaum von *ibaPDA-Client* und *ibaAnalyzer*.

Zeitbasis

Alle Signale dieses Moduls werden mit dieser Zeitbasis erfasst.

Modulname als Präfix verwenden

Diese Option stellt den Modulnamen den Signalnamen voran.

Diagnose

Ausführliche Protokollierung

Wenn die ausführliche Protokollierung deaktiviert ist (False), werden nur Fehler protokolliert.

Wenn diese aktiviert ist (True), werden alle Abfrage- und Antwortdetails protokolliert.

Trigger

Aktualisierungszeit

Je nach Sendemodus definiert die Aktualisierungszeit die Zeitspanne zwischen zwei aufeinanderfolgenden Abfragen.

- *Zyklisch*: Eine Abfrage wird in regelmäßigen Abständen entsprechend der Aktualisierungszeit gesendet.
- *Bei Änderung/Bei Trigger*: Die Aktualisierungszeit dient als Entprellzeit. Wenn zwischen der letzten Abfrage und dem neuen Trigger weniger Zeit verstrichen ist, als in der Aktualisierungszeit definiert wurde, wird die Abfrage ignoriert.

Sendemodus

Damit wird festgelegt, wann eine neue HTTP(S)-Abfrage gesendet wird.

- *Zyklisch*: Eine Abfrage wird in regelmäßigen Abständen entsprechend der Aktualisierungszeit gesendet.
- *Bei Änderung*: Eine Abfrage wird jedes Mal gesendet, wenn sich die Daten eines Triggersignals ändern.
- *Bei Trigger*: Eine Abfrage wird jedes Mal gesendet, wenn eine steigende Flanke am Triggersignal erkannt wird.

Triggersignal

Das Feld *Triggersignal* erscheint, wenn der Sendemodus *Bei Änderung* oder *Bei Trigger* ausgewählt ist. Das Triggersignal bestimmt, wann eine neue HTTP(S)-Abfrage gesendet wird.

Im Sendemodus *Bei Änderung* können Sie ein analoges, digitales oder Textsignal auswählen. Im Sendemodus *Bei Trigger* können Sie ein digitales Signal auswählen.

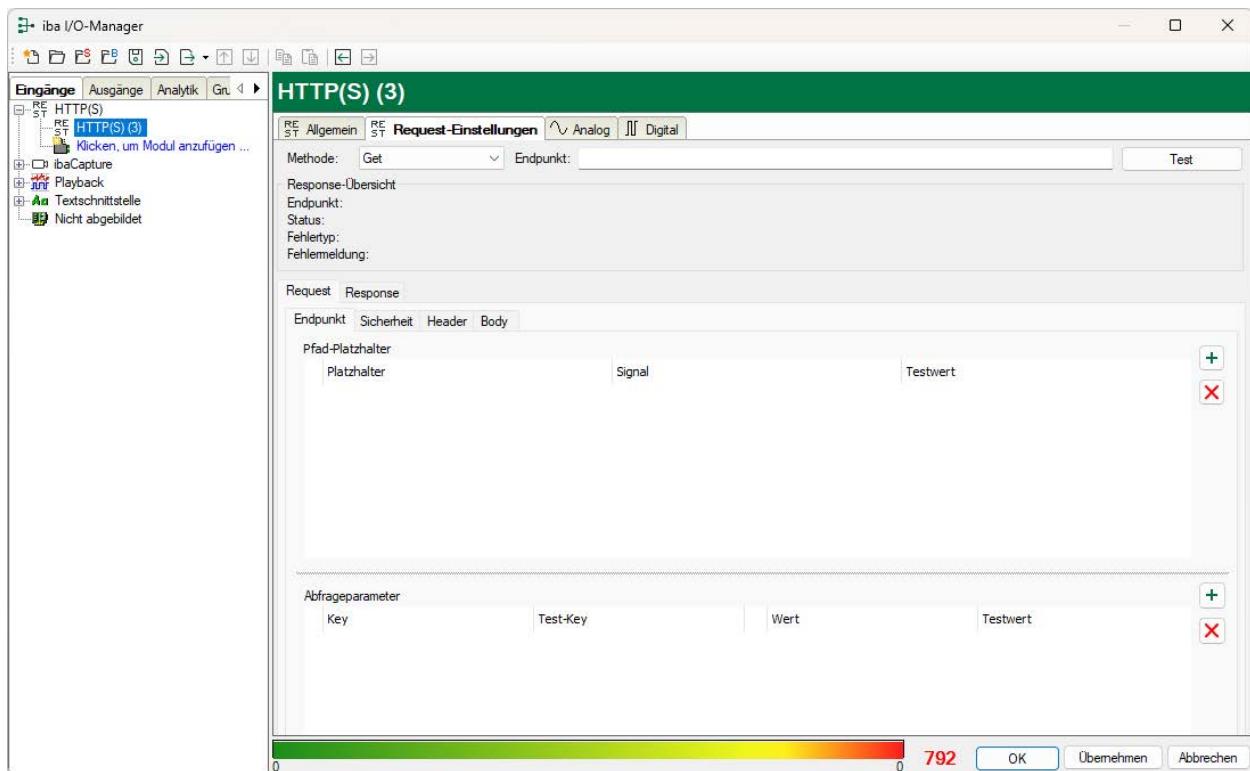
Verarbeitung

Dezimalpunkt

Konfigurieren Sie, welches Zeichen als Dezimalpunkt verwendet wird (Punkt oder Komma).

3.2.4 Request-Einstellungen

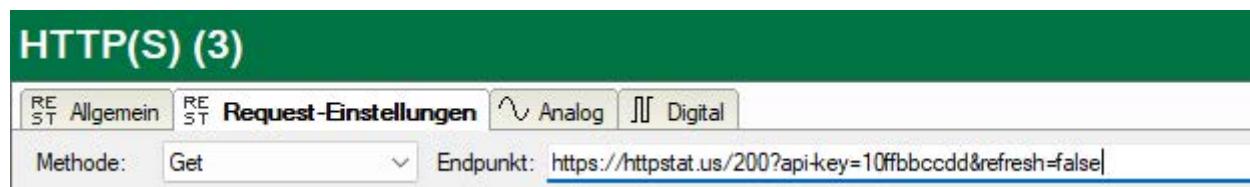
Im Register *Request-Einstellungen* konfigurieren Sie die Einstellungen für eine Abfrage.



Endpunkt

Im Textfeld *Endpunkt* definieren Sie die URL für die Abfrage. In der Auswahlliste *Methode* wählen Sie die HTTP-Methode aus. Das HTTP(S)-Modul unterstützt alle Standard-HTTP-Methoden.

Ein Endpunkt besteht aus dem Pfad und der Abfrage, getrennt durch "?".



Beispiel:

GET https://httpstat.us/200?api-key=10ffbbccdd&refresh=false

Methode: GET

Pfad: https://httpstat.us/200

Abfrage: api-key=10ffbbccdd&refresh=false

3.2.4.1 Request – Register Endpunkt

Bei der Definition des Endpunkts können Sie Abfrageparameter und Pfad-Platzhalter verwenden. Im Register *Endpunkt* werden die Abfrageparameter aufgelistet und die Pfad-Platzhalter definiert.

Ein Endpunkt kann hier direkt per Copy & Paste eingefügt werden. Dabei werden die mit der Zeichenfolge ?= beginnenden Parameter automatisch als Abfrageparameter angelegt.

Die Zeichenfolge ?= kann NICHT händisch eingegeben werden. Hierfür muss ein Abfrageparameter angelegt werden.

3.2.4.1.1 Request-Settings – Abfrageparameter

Die Abfrageparameter werden in einer Tabelle aufgelistet. Ein Abfrageparameter besteht immer aus einem Key (Schlüssel) und einem Wert.

Sie können einen neuen Abfrageparameter über den <+>-Button neben der Tabelle hinzufügen. Mit dem <x>-Button wird ein markierter Abfrageparameter gelöscht.

Dem Beispiel <https://httpstat.us/200?api-key=10ffbbccdd&refresh=false> folgend, werden in der Tabelle der Abfrageparameter zwei Schlüssel-Wert-Paare aufgelistet.

| Abfrageparameter | | | |
|---|------------|--------------|----------|
| Key | Test-Key | Wert | Testwert |
| ▶ <input checked="" type="checkbox"/> api-key | ▼ Test-Key | ✓ 10ffbbccdd | |
| ✓ refresh | | ✓ false | |

Key

Der Schlüssel des Abfrageparameter kann ein statischer Wert oder ein Signal sein.

Test-Key

Wenn für *Key* ein Signal ausgewählt ist, wird der Wert in der Spalte *Test-Key* als Schlüssel für den Abfrageparameter beim Testen der Abfrage verwendet. Andernfalls ist die Spalte nicht sichtbar.

Wert

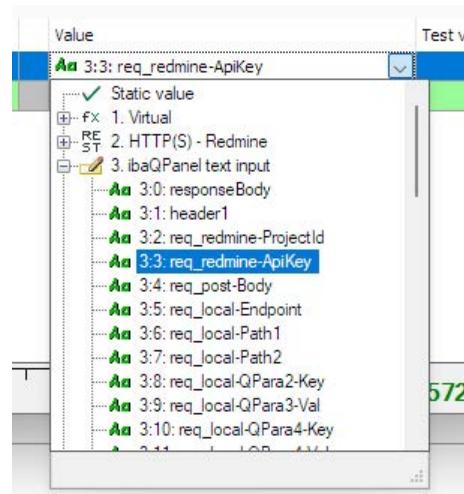
Der Wert des Abfrageparameter kann ein statischer Wert oder ein Signal sein.

Testwert

Wenn für *Wert* ein Signal ausgewählt ist, wird der Wert in der Spalte *Testwert* als Wert für den Abfrageparameter beim Testen der Abfrage verwendet. Andernfalls ist die Spalte nicht sichtbar.

Wenn die Werte in der Tabelle geändert werden, wird der Eintrag im Textfeld *Endpunkt* automatisch aktualisiert und umgekehrt.

Im nachfolgenden Beispiel wird ein Signal als Wert für den Abfrageparameter ausgewählt.



Der vollständige Endpunkt der aktuellen Konfiguration lautet wie folgt:

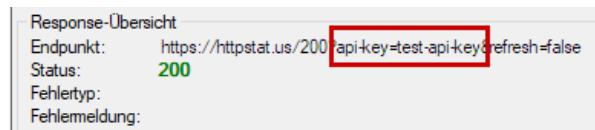
[https://httpstat.us/200?api-key={\[3:3\]}&refresh=false](https://httpstat.us/200?api-key={[3:3]}&refresh=false)

Wird die Abfrage bei laufender Erfassung gesendet, wird die Signal-ID {[3:3]} durch den aktuellen Wert des Signals ersetzt.

Wenn beispielsweise das Signal {[3:3]} den aktuellen Wert *test* hat, ergibt sich für den Endpunkt
<https://httpstat.us/200?api-key=test&refresh=false>.

Wenn Sie die Abfrage testen, ist der Wert des Signals jedoch nicht verfügbar. Sie können deshalb in der Spalte *Testwert* einen Wert für den Test definieren.

Nachdem Sie den Button <Test> gedrückt haben, zeigt die Response-Übersicht, dass der Endpunkt den Wert des api-key-Parameters {[3:3]} in den definierten Testwert (test-api-key) aufgelöst hat.



3.2.4.1.2 Request-Settings – Pfad-Platzhalter

Pfad-Platzhalter werden verwendet, um einen Teil des Pfades dynamisch ändern zu können. In diesen Fällen wird kein statischer Wert gesendet, sondern der aktuelle Wert des definierten Signals. Statische Werte müssen direkt im Endpunkt-Textfeld definiert werden.

HTTP(S) (3)

RE ST Allgemein RE ST Request-Einstellungen Analog Digital

Methode: Get Endpunkt: https://httpstat.us/200?api-key=10ffbbccdd&refresh=false&key3=value3 Test

Response-Übersicht
Endpunkt:
Status:
Fehlertyp:
Fehlernmeldung:

Request Response

Endpunkt Sicherheit Header Body

| Pfad-Platzhalter | Platzhalter | Signal | Testwert |
|------------------|-------------|--------|---|
| | | | + X |

Ereignisparameter

| Key | Test-Key | Wert | Testwert |
|--|----------|---|----------|
| ► ✓ api-key | | ✓ 10ffbbccdd | |
| ✓ refresh | | ✓ false | |
| ✓ key3 | | ✓ value3 | |

Sie können einen neuen Pfad-Platzhalter mit dem <+>-Button neben der Tabelle hinzufügen. Mit dem <x>-Button wird ein markierter Pfad-Platzhalter gelöscht.

Ein Pfad-Platzhalter besteht aus dem Platzhalter, dem Signal und dem Testwert.

Platzhalter

Ein Platzhalter wird automatisch generiert. Die Bezeichnung kann bei Bedarf geändert werden. Die Bezeichnung für den Platzhalter muss eindeutig sein und mit einer geschweiften Klammer beginnen und enden.

Signal

Wählen Sie hier das Signal, dessen Wert den Platzhalter ersetzt.

Testwert

Definieren Sie hier einen Testwert, der beim Testen der Abfrage verwendet wird.

Im nachfolgenden Beispiel ist für den Platzhalter {1} das Signal [3:6] ausgewählt und der Testwert mit 200 definiert.

| Platzhalter | Signal | Testwert |
|-------------|--|----------|
| ► {1} | Aa 3:6: req_local-Path1 | 200 |

Entsprechend dem Beispiel wird das *Endpunkt*-Textfeld automatisch aktualisiert.

Methode: Get Endpunkt: https://httpstat.us/{1}?api-key=[3:3]&refresh=false

Wenn die Abfrage gesendet wird, wird {1} durch den Testwert 200 ersetzt.

Response-Übersicht
Endpunkt: https://httpstat.us/200?api-key=test-api-key&refresh=false
Status: 200
Fehlertyp:
Fehlmeldung:

Anstelle von {1} können Sie auch eine ausführliche Bezeichnung verwenden. (z. B. {statusCode}).

Beispiel:

| | | |
|-------------------------------|----------------------------|----------|
| Pfad-Platzhalter | Signal | Testwert |
| Platzhalter ► {statusCode} | Aa 3:5: req_local-Endpoint | v 200 |

Endpunkt: https://httpstat.us/{statusCode}?api-key=[3:3]&refresh=false

Hinweis



Ein Platzhalter kann im *Endpunkt*-Textfeld verschoben werden. Markieren Sie hierfür den Platzhalter, schneiden ihn mit <STRG> + <X> aus und fügen ihn an anderer Stelle mit <STRG> + <V> wieder ein.

3.2.4.2 Request – Register Sicherheit

Im Register *Sicherheit* nehmen Sie sicherheitsrelevante Einstellungen vor.

| Request Response | |
|--|-------------------|
| Endpunkt | Sicherheit |
| <input type="checkbox"/> Automatische Umleitung zulassen | |
| Server-Authentifizierung | |
| TLS/SSL-Validierung: | Windows |
| Client-Authentifizierung | |
| <input type="checkbox"/> Client-Zertifikat: | X Kein Zertifikat |
| <input type="checkbox"/> Authentifizierung verwenden | |

Automatische Umleitung zulassen

Wenn das Ziel mit einer Umleitung antwortet (HTTP-Statuscode 300 bis 399), wird die Umleitung verfolgt, wenn diese Option aktiviert ist.

Server-Authentifizierung

TLS/SSL-Validierung

Bei der Verbindung zu einem HTTPS-Endpunkt werden die SSL-Zertifikate überprüft. Wählen Sie hier das Verhalten für die Zertifikatsvalidierung.

- **Keine:** Das SSL-Zertifikat wird nicht validiert. Diese Option ermöglicht unsichere Verbindungen und wird nicht empfohlen.
- **Windows:** Das SSL-Zertifikat wird automatisch vom Betriebssystem validiert.
- **Zertifikatspeicher:** Das SSL-Zertifikat wird durch den ibaPDA-Zertifikatspeicher validiert.

Client-Authentifizierung

Client-Zertifikat

Wenn diese Option aktiviert ist, wird das hier ausgewählte Zertifikat als Client-Zertifikat mit der Abfrage gesendet.

Authentifizierung verwenden

Wenn diese Option aktiviert ist, wird die Abfrage mit der konfigurierten Authentifizierung gesendet.

ibaPDA unterstützt unterschiedliche Authentifizierungstypen:

- Basic
- Json-Web-Token
- OAuth2.0 (ROPC und Client-Anmeldeinformationen)

Wählen Sie den gewünschten Authentifizierungstypen in der Auswahlliste *Typ* aus. Die folgenden Konfigurationsoptionen ändern sich entsprechend.

Basic

Mit der Basic-Authentifizierung können Sie einen Benutzernamen und ein Kennwort definieren. Die Authentifizierung wird als Authorization-Header zur Abfrage hinzugefügt.

| | |
|-------------------|----------|
| Einstellungen | |
| Typ: | Basic |
| Hinzufügen: | Header |
| Abfrageparameter: | |
| Basic | |
| Benutzername: | testUser |
| Kennwort: | ***** |

JSON Web Token (JWT)

Wählen Sie den Typ für die JWT-Authentifizierung aus:

- Header: Das JSON-Web-Token wird im Authorization-Header als Inhaber-Token gesendet.
- Query: Das JSON-Web-Token wird als Abfrageparameter in der URL mit dem angegebenen Schlüssel gesendet. Geben Sie zusätzlich den Abfrageparameter ein.

| | |
|----------------------|----------------------|
| Einstellungen | |
| Typ: | Json Web Token (JWT) |
| Hinzufügen: | Header |
| Abfrageparameter: | my-auth-key |
| Json Web Token (JWT) | |
| Algorithm: | HS256 |
| Secret: | |
| Payload: | |

| | |
|----------------------|----------------------|
| Einstellungen | |
| Typ: | Json Web Token (JWT) |
| Hinzufügen: | Query |
| Abfrageparameter: | my-auth-key |
| Json Web Token (JWT) | |
| Algorithm: | HS256 |
| Secret: | |
| Payload: | |

Weitere Parameter für beide Typen:

Algorithmus

Der Algorithmus wird für die Erzeugung der Signatur verwendet. Derzeit wird nur HS256 (HMAC mit SHA256)unterstützt.

Secret

Das "Secret" (geheimer Schlüssel) wird als Schlüssel für den HS256-Algorithmus verwendet.

Payload

Die "Payload" (Nutzdaten) enthalten benutzerdefinierte Informationen, die zur Validierung der Abfrage verwendet werden.

OAuth2.0

Das OAuth2.0-Protokoll definiert mehrere Autorisierungsstandards. *ibaPDA* unterstützt ROPC (Benutzername/Kennwort) und den Grant-Typen *Client Credentials*.

"Dreistufige" Grant-Typen werden nicht unterstützt.

Hinweis



Abfragen werden automatisch mit den angegebenen Informationen autorisiert. Nachfolgende Abfragen verwenden das ausgestellte Token. Wenn das Token abgelaufen ist und ein Refresh-Token verfügbar ist, wird das Token mit den angegebenen Informationen aufgefrischt.

Wählen Sie den gewünschten Grant-Typen aus der Auswahlliste:

- Password
- Client credentials

Einige allgemeine Eigenschaften gelten für beide Grant-Typen:

Zugriffstoken-URL

Definiert den Endpunkt zum Abrufen und Aktualisieren des Tokens.

Geltungsbereich

Definiert optional den gewünschten Geltungsbereich des Tokens. Eine durch Leerzeichen getrennte Liste von Bereichen oder Berechtigungen, die *ibaPDA* benötigt.

Client-ID

Die Anwendungs-ID (Client-ID), die *ibaPDA* zugewiesen ist.

Client-Secret

Definiert optional das Client-Secret.

Client-Authentifizierung senden in

Legt fest, ob die Authentifizierung für Token-bezogene Abfragen im Header oder im Body gesendet wird.

Grant-Typ Password:

Neben den allgemeinen Eigenschaften erlaubt der Grant-Typ *Password* das Hinzufügen eines Benutzernamens und eines Kennworts.

The screenshot shows two side-by-side configuration panels. On the left is the 'Einstellungen' (Settings) panel, which has 'Typ: OAuth 2.0' and 'Hinzufügen: Header' selected. On the right is the 'OAuth 2.0' panel, where 'Grant Type:' is set to 'Password'. Other fields include 'Zugriffstoken-URL: https://api.test.com/oauth/token', 'Scope: read_user', 'Client-ID: 110ddfe', 'Client secret: *****', and 'Client-Authentifizierung senden in: Body' (Body radio button selected). Below these are fields for 'Benutzername: testUser' and 'Kennwort: *****'.

Grant-Typ Client credentials

Hier gelten die allgemeinen Eigenschaften, ohne Kennwort und Benutzername.

This screenshot shows the same two-panel configuration as above, but with 'Grant Type:' set to 'Client credentials'. The right panel's fields remain identical: 'Zugriffstoken-URL: https://api.test.com/oauth/token', 'Scope: read_user', 'Client-ID: 100ddfe', 'Client secret: *****', and 'Client-Authentifizierung senden in: Body' (Body radio button selected).

3.2.4.3 Request – Register Header

Im Register *Header* werden die Abfrage-Header definiert.

The screenshot shows the 'Request' tab selected. Under the 'Header' tab, there is a table titled 'Request-Header' with columns 'Key', 'Test-Key', 'Wert', and 'Testwert'. A '+' button is at the top right for adding new entries, and a '-' button is at the bottom right for deleting selected ones.

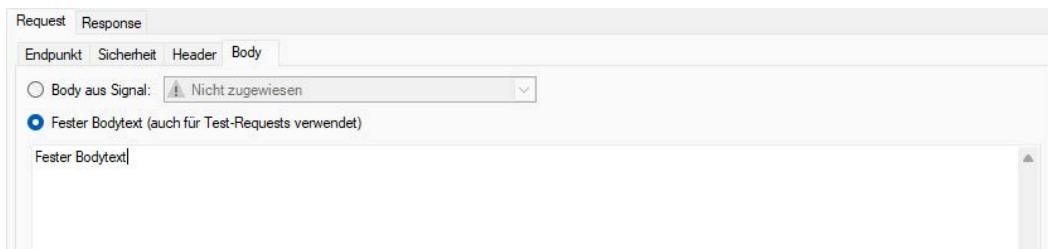
Sie können für den Key (Schlüssel) und den Wert jeweils einen statischen Wert eingeben oder ein Signal auswählen.

Mit dem <+>-Button fügen Sie einen neuen Request-Header hinzu, mit dem <x>-Button können Sie einen markierten Request-Header löschen.

3.2.4.4 Request – Register Body

Der Body einer HTTP-Abfrage enthält alle Informationen, die an den Webserver gesendet werden. Dies kann entweder ein fester Text oder ein Signal sein. Wählen Sie die gewünschte Option aus.

Wenn Sie *Fester Bodytext* ausgewählt haben, können Sie den Text im Textfeld darunter eingeben.



Hinweis

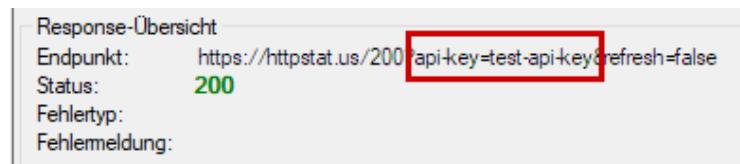


Auch wenn die Option *Body aus Signal* ausgewählt ist, bleibt das Texteingabefeld aktiv, da der Text im Textfeld als Testwert zum Testen der Abfrage verwendet werden kann.

3.2.5 Register Response

Die Einstellungen im Register *Response* beziehen sich auf die Antwort, die auf eine Abfrage folgt.

Nach dem Test einer Abfrage, gibt die Response-Übersicht eine Übersicht über die Antwort. Die Übersicht enthält den aufgelösten Endpunkt, den Statuscode, den Fehlertyp und die Fehlermeldung.



Wenn die Abfrage fehlgeschlagen ist, erscheint eine Fehlermeldung. Der Fehler kann dann anhand des Fehlertyps und der Fehlermeldung identifiziert werden.

Je nach gewählter Authentifizierung zeigt die Response-Übersicht auch Fehler an, die während des Authentifizierungsprozesses aufgetreten sind.

Mögliche Fehlermeldungen:

| Fehlertyp | Ursache |
|--------------------------|--|
| RequestFailed | Die Abfrage ist gescheitert. Keine weiteren Informationen verfügbar. |
| SslVerificationFailed | Beim Aufbau einer SSL-Verbindung ist ein Fehler aufgetreten. |
| StatusCodesNotSuccessful | Der HTTP-Statuscode ist nicht erfolgreich (4xx). |
| Timeout | Der Server hat nicht rechtzeitig geantwortet. |

| FehlerTyp | Ursache |
|------------------------------------|---|
| InvalidEndpoint | Die URL ist nicht richtig formatiert. |
| IllegalPlaceholderCharacter | Der Wert eines Platzhalters enthält unzulässige Zeichen. |
| OAuth2MissingAuthorizationEndpoint | Betrifft nur OAuth2: Die URL des Zugriffstokens fehlt |
| OAuth2InvalidAuthorizationResponse | Betrifft nur OAuth2: Die an den Autorisierungsserver gesendete Abfrage war nicht erfolgreich. |
| OAuth2PasswordUsername-Missing | Betrifft nur OAuth2: ROPC Der Benutzername oder das Kennwort fehlt. |
| OAuth2InvalidRequest | Betrifft nur OAuth2: Die Abfrage ist ungültig. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("invalid_request") |
| OAuth2InvalidClient | Betrifft nur OAuth2: Die Client-Authentifizierung ist fehlgeschlagen. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("invalid_client") |
| OAuth2InvalidGrant | Betrifft nur OAuth2: Die Berechtigungserteilung (Grant) ist ungültig. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("invalid_grant") |
| OAuth2UnauthorizedClient | Betrifft nur OAuth2: Der Client ist nicht berechtigt, den Grant anzufordern. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("unauthorized_client") |
| OAuth2UnsupportedGrantType | Betrifft nur OAuth2: Der angeforderte Grant wird nicht unterstützt. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("unsupported_grant") |
| OAuth2InvalidScope | Betrifft nur OAuth2: Der Abfragebereich ist nicht gültig. Weitere Details siehe https://datatracker.ietf.org/doc/html/rfc6749#section-5.2 ("invalid_scope") |
| InvalidClientCertificate | Das ausgewählte Client-Zertifikat ist nicht gültig. |
| ServerCertificateInvalidDate | Das Datum des SSL-Zertifikats liegt nicht in einem gültigen Bereich. |
| ServerCertificateNotTrusted | Das SSL-Zertifikat ist im Zertifikatsspeicher vorhanden, aber nicht vertrauenswürdig. |
| ServerCertificateInvalid | Das SSL-Zertifikat ist ungültig. |
| ServerCertificateChain-NotTrusted | Die SSL-Zertifikatskette ist nicht vertrauenswürdig. |
| ServerCertificateChainNotValid | Die SSL-Zertifikatskette ist ungültig. |
| ServerCertificateMissingInManager | Das SSL-Zertifikat fehlt im Zertifikatsspeicher. |

3.2.5.1 Response – Register Allgemein

Im Bereich *Response-Attribute* werden Informationen über die Antwort angezeigt, wie Dauer, Statuscode, Fehlerart, Fehlermeldung und Endpunkt.

| Request Response | |
|-----------------------|------|
| Allgemein Header Body | |
| Response-Attribute | |
| Attribut | Wert |
| ► Duration | |
| StatusCode | |
| ErrorType | |
| ErrorMessage | |
| Endpoint | |

Das HTTP(S)-Modul enthält auch automatisch generierte Signale für diese Attribute. Die Signale finden Sie im Register *Analog* des Moduls.

| HTTP(S) (3) | | | | |
|-------------|---------------------------|----------|-----------------------|-------------------------------------|
| RE ST | Allgemein | RE ST | Request-Einstellungen | Analoga |
| | Name | | Einheit | Aktiv Istwert |
| 0 | Statuscode der Antwort | | | <input checked="" type="checkbox"/> |
| 1 | Dauer der Antwort | | | <input checked="" type="checkbox"/> |
| 2 | Fehlerart der Antwort | | | <input checked="" type="checkbox"/> |
| 3 | Fehlermeldung der Antwort | | | <input checked="" type="checkbox"/> |
| 4 | Response-Body | | | <input checked="" type="checkbox"/> |
| 5 | Endpunkt | | | <input checked="" type="checkbox"/> |

3.2.5.2 Response – Register Header

Die Tabelle *Response-Header* enthält die in der Antwort empfangenen Header.

| Request Response | |
|--|--|
| Allgemein Header Body | |
| Header-Signale hinzufügen Header-Signale entfernen | |
| Response-Header | |
| Key | Wert |
| ► Access-Control-Allow-Credentials | true |
| Access-Control-Allow-Methods | GET, POST |
| Access-Control-Allow-Origin | * |
| Connection | keep-alive |
| Date | Tue, 16 Apr 2024 09:29:12 GMT |
| Server | openresty |
| X-Cache-Key | /data/2.5/weather?lang=de&lat=49.33&lon=12.18&units=metric |

Mit den Buttons *<Header-Signale hinzufügen>* und *<Header-Signale entfernen>* können Sie die Header zu den Signalen des Moduls hinzufügen oder entfernen. Wenn die Signale hinzugefügt sind, erscheinen sie im Register *Analog*.

| HTTP(S) - Weather (6) | | | |
|-----------------------|----------------------------------|------|--------|
| | Name | Unit | Active |
| 0 | Response status code | | ✓ |
| 1 | Response duration | | ✓ |
| 2 | Response error type | | ✓ |
| 3 | Response error message | | ✓ |
| 4 | Response body | | ✓ |
| 5 | Connection | | ✓ |
| 6 | Access-Control-Allow-Credentials | | ✓ |
| 7 | Access-Control-Allow-Methods | | ✓ |
| 8 | Access-Control-Allow-Origin | | ✓ |
| 9 | Connection | | ✓ |
| 10 | Date | | ✓ |
| 11 | Server | | ✓ |
| 12 | X-Cache-Key | | ✓ |

3.2.5.3 Response – Register Body

Im Register *Response – Body* werden die empfangenen Daten angezeigt.

| Request | | |
|---------|---------|--|
| General | Headers | Body |
| | | <p>Body content:</p> <pre>{ "coord": { "lat": 52.3687, "lon": 10.5033 }, "weather": [{ "id": 804, "main": "Clouds", "description": "Bedeckt", "icon": "04d" }], "base": "stations", "main": { "temp": 26.62, "feels_like": 26.62, "temp_min": 25.01, "temp_max": 28.62, "pressure": 1010, "humidity": 34, "sea_level": 1010, "gmd_level": 962 }, "visibility": 10000 }</pre> |

Die empfangenen Daten sind auch als automatisch generiertes Signal *Response Body* im Register *Analog* verfügbar.

| HTTP(S) - Weather (6) | | | |
|-----------------------|----------------------------------|------|--------|
| | Name | Unit | Active |
| 0 | Response status code | | ✓ |
| 1 | Response duration | | ✓ |
| 2 | Response error type | | ✓ |
| 3 | Response error message | | ✓ |
| 4 | Response body | | ✓ |
| 5 | Connection | | ✓ |
| 6 | Access-Control-Allow-Credentials | | ✓ |
| 7 | Access-Control-Allow-Methods | | ✓ |
| 8 | Access-Control-Allow-Origin | | ✓ |
| 9 | Connection | | ✓ |
| 10 | Date | | ✓ |
| 11 | Server | | ✓ |
| 12 | X-Cache-Key | | ✓ |

Die abgefragten Daten befinden sich als Wert im Signal *Response Body*. Mit Hilfe eines Texttrenner-Moduls können Sie die empfangenen Daten zerlegen und in übersichtlicher Form darstellen. Siehe auch Anwendungsbeispiel, Kapitel **Anwendungsbeispiel HTTP(S)**, Seite 25.

3.2.6 Analoge und digitale Signale

Das HTTP(S)-Modul enthält automatisch generierte Signale.

Register Analog

Standardmäßig enthält jedes HTTP(S)-Modul diese analogen Signale im Register *Analog*:

- Statuscode der Antwort
- Dauer der Antwort
- Fehlerart der Antwort
- Fehlermeldung der Antwort
- Response-Body
- Endpunkt

Die Signale zeigen immer die Werte der letzten Abfrage.

| HTTP(S) (3) | | | | |
|-------------|---------------------------|---------|-------------------------------------|---------|
| | Name | Einheit | Aktiv | Istwert |
| 0 | Statuscode der Antwort | | <input checked="" type="checkbox"/> | |
| 1 | Dauer der Antwort | | <input checked="" type="checkbox"/> | |
| 2 | Fehlerart der Antwort | | <input checked="" type="checkbox"/> | |
| 3 | Fehlermeldung der Antwort | | <input checked="" type="checkbox"/> | |
| 4 | Response-Body | | <input checked="" type="checkbox"/> | |
| 5 | Endpunkt | | <input checked="" type="checkbox"/> | |

Register Digital

Jedes HTTP(S)-Modul liefert das digitale Signal *Ausstehende Abfrage*. Dieses Signal zeigt an, ob eine Abfrage gerade aktiv ist (1) oder nicht (0).

| HTTP(S) (3) | | | |
|-------------|---------------------|-------------------------------------|---------|
| | Name | Aktiv | Istwert |
| 0 | Ausstehender Antrag | <input checked="" type="checkbox"/> | 1 |

4 Anwendungsbeispiel HTTP(S)

Aufgabe:

Von einem Preisportal eines Energieversorgers sollen die Energiepreise für einen Tag in stündlichen Werten abgefragt werden.

4.1 Vorbereitung

Für die Datenabfrage wurden zuvor die erforderlichen Parameter und zugelassenen Methoden beim Portalbetreiber in Erfahrung gebracht. Auf dieser Basis wurde eine HTTP(S)-Abfrage in java script formuliert. Die Abfrage liefert die gewünschten Daten.

Die Aufgabe ist nun, in *ibaPDA* ein HTTP(S)-Modul so zu konfigurieren, dass die Abfrage auch damit funktioniert.

Beispielabfrage in java script

```
const url = https://seffaflik.epias.com.tr/electricity-service/v1/markets/dam/data/mcp;  
const payload = {  
    startDate: "2024-04-30T00:00:00+03:00",  
    endDate: "2024-04-30T00:00:00+03:00",  
    page: {  
        number: 1,  
        size: 24,  
        sort: {  
            direction: "ASC",  
            field: "date"  
        }  
    }  
};  
  
fetch(url, {  
    method: 'POST',  
    headers: {  
        'Content-Type': 'application/json'  
    },  
    body: JSON.stringify(payload)  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error('Error:', error));
```

4.2 Konfiguration des ibaPDA HTTP(S)-Moduls

Die Abfrage konfigurieren Sie im Register *Request-Einstellungen*.

Die nachfolgende Abbildung zeigt am Beispiel der java script-Abfrage die Entsprechungen im HTTP(S)-Modul.

```

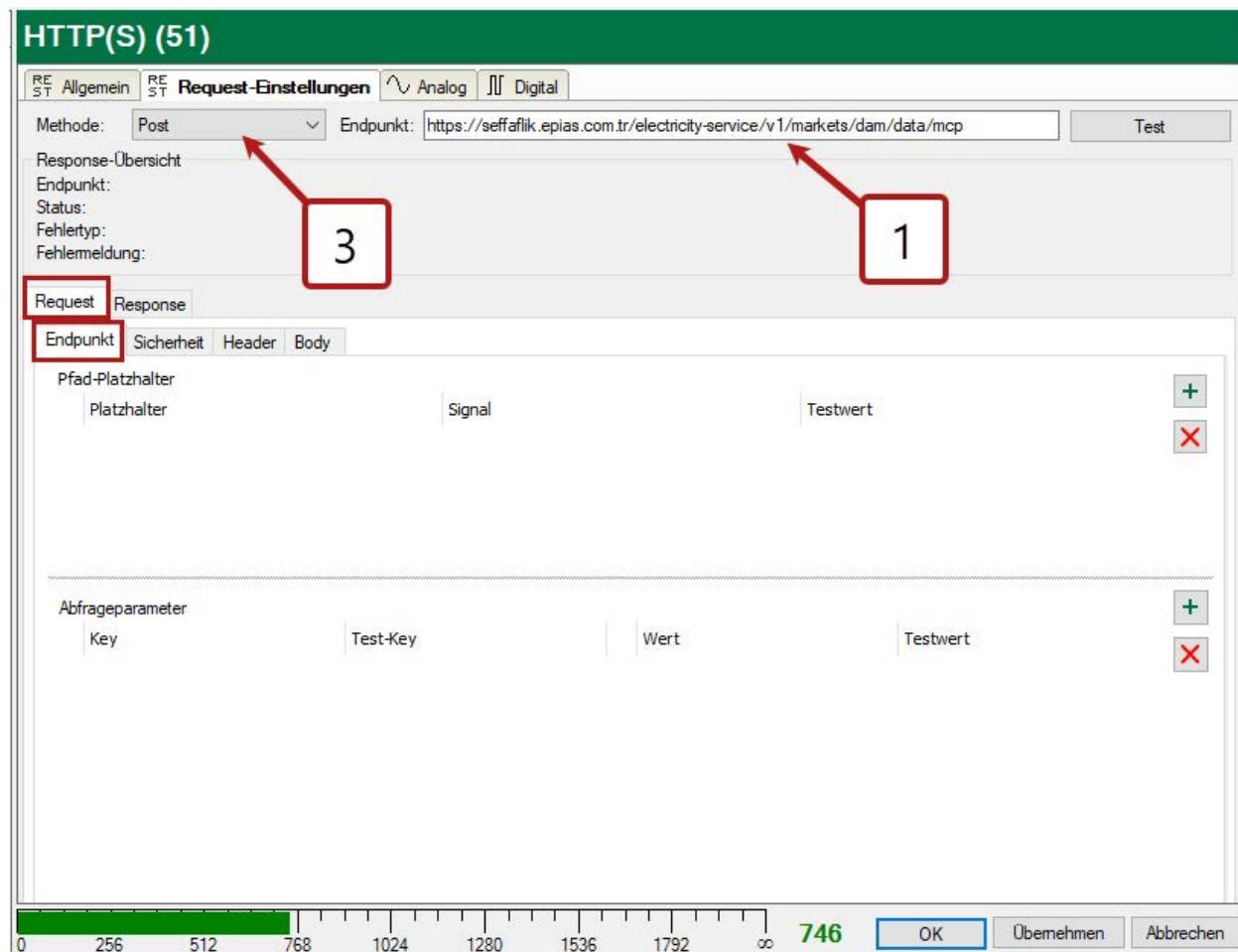
1 const url = https://seffaflik.epias.com.tr/electricity-service/v1/markets/dam/data/mcp;
2 const payload = {
3     startDate: "2024-04-30T00:00:00+03:00",
4     endDate: "2024-04-30T00:00:00+03:00",
5     page: {
6         number: 1,
7         size: 24,
8         sort: {
9             direction: "ASC",
10            field: "date"
11        }
12    }
13 };
14
15 fetch(url, {
16     method: 'POST',
17     headers: {
18         'Content-Type': 'application/json'
19     },
20     body: JSON.stringify(payload)
21 })
22 .then(response => response.json())
23 .then(data => console.log(data))
24 .catch(error => console.error('Error:', error));

```

| | |
|---|--|
| 1 | Endpunkt |
| 2 | Request Body (bei Verwendung im HTTP(S)-Modul im json-Format schreiben, s. u.) |
| 3 | Methode |
| 4 | Request Header |
| 5 | Ausgabe der Antwort (Response) |

Dementsprechend sehen die Eintragungen im Modul wie folgt aus:

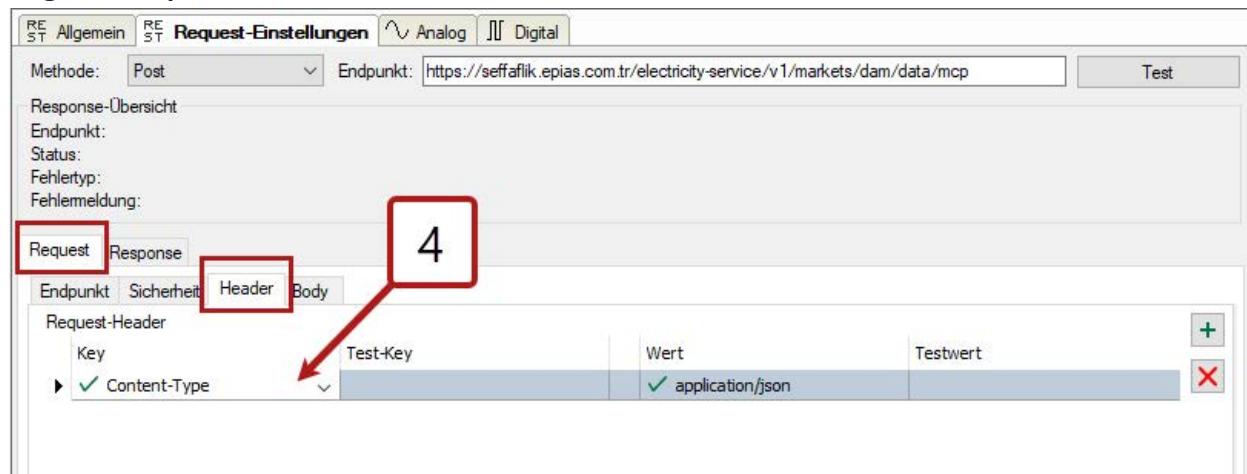
Register Request – Einstellungen



Pfad-Platzhalter sind hier nicht erforderlich, da in der Angabe zum Endpunkt schon der absolute Pfad vollständig eingetragen ist.

Abfrageparameter sind hier nicht erforderlich. Die Abfrage wird im Register *Body* vollständig formuliert.

Register Request – Header



Das Key-Value-Paar "Content-Type" spezifiziert hier, dass die Abfrage als json-Script formuliert ist.

Register Request – Body

The screenshot shows the 'Request' tab selected. Below it, there are tabs for 'Endpunkt', 'Sicherheit', 'Header', and 'Body'. The 'Body' tab is highlighted with a red box. A red arrow points from the number '2' to the JSON body content.

```
{
  "startDate": "2024-04-30T00:00:00+03:00",
  "endDate": "2024-04-30T00:00:00+03:00",
  "page": {
    "number": 1,
    "size": 24,
    "sort": {
      "direction": "ASC",
      "field": "date"
    }
  }
}
```

Im Gegensatz zur java script-Version der Abfrage aus dem Beispiel oben muss der Body für das HTTP(S)-Modul bereits im json-Format formuliert werden.

Die Abfrage fordert hier gezielt die Daten von 24 Stunden am 30.04.2024 in aufsteigender Reihenfolge (Datum/Uhrzeit) an.

Abfrage testen

Sind die Eingaben vollständig, kann mit Klick auf den <Test>-Button eine Probeabfrage gestartet werden.

Wenn die Abfrage erfolgreich war, erscheint bei Status der Wert **200**.

The screenshot shows the 'Response' tab selected. Above it, there are tabs for 'Allgemein', 'Header', and 'Body'. A red box highlights the 'Test' button. A red arrow points from the number '200' to the 'StatusCode' entry in the 'Response-Attribute' table.

| Attribut | Wert |
|--------------|---|
| Duration | 00:00:00.379000 |
| StatusCode | 200 |
| ErrorType | None |
| ErrorMessage | |
| Endpoint | https://seffaflik.epias.com.tr/electricity-service/v1/markets/dam/data/mcp |

Wenn entsprechende Daten auf dem Zielsystem vorhanden sind, werden diese im Register *Response – Body* angezeigt.

The screenshot shows the 'Request-Einstellungen' tab selected. The 'Methode' dropdown is set to 'Post'. The 'Endpunkt' field contains the URL <https://seffaflik.epias.com.tr/electricity-service/v1/markets/dam/data/mcp>. The 'Status' field shows '200'. Below the endpoint, there are fields for 'FehlerTyp:' and 'Fehlmeldung:', both currently empty.

The 'Response' tab is selected, and the 'Body' tab is highlighted with a red box. The 'Body-Inhalt:' section displays the following JSON data:

```
{
  "items": [
    {
      "date": "2024-04-30T00:00:00\u002B03:00",
      "hour": "00:00",
      "price": 2399.989999999998,
      "priceUsd": 73.95,
      "priceEur": 68.91
    },
    {
      "date": "2024-04-30T01:00:00\u002B03:00",
      "hour": "01:00",
      "price": 2363.139999999999,
      "priceUsd": 72.82,
      "priceEur": 67.86
    },
    {
      "date": "2024-04-30T02:00:00\u002B03:00",
      "hour": "02:00",
      "price": 1631.98,
      "priceUsd": 50.29,
      "priceEur": 46.86
    },
    {
      "date": "2024-04-30T03:00:00\u002B03:00",
      "hour": "03:00",
      "price": 1631.98,
      "priceUsd": 50.29,
      "priceEur": 46.86
    },
    {
      "date": "2024-04-30T04:00:00\u002B03:00",
      "hour": "04:00",
      "price": 1631.98,
      "priceUsd": 50.29,
      "priceEur": 46.86
    }
  ]
}
```

Die Antwort liefert 24 Datensätze mit vollständigem Datum, die volle Stunde des Tages und die Preisinformationen als Angaben in Landeswährung, USD und EUR.

Hinweis



Dieses Beispiel ist sehr einfach, weil es nur die Daten eines festen Tages abruft. Realistischer wäre eine variable Abfrage, in der das Datum durch Signale dynamisch vorgegeben werden kann. Hierzu könnten Sie z. B. mithilfe eines Texterzeuger-Moduls den Request-Body unter Nutzung von Textsignalen dynamisch gestalten und dann im Register *Request – Body* die Option *Body aus Signal anwählen*. Das Signal wäre dann ein Textsignal, das den vollständigen Body enthält.

4.3 Weiterverarbeitung der abgefragten Daten

Die wichtigsten Informationen stehen im Register *Analog* des Moduls zur Verfügung.

| Name | Einheit | Aktiv | Istwert |
|--------------------------|---------|-------------------------------------|---|
| 0 Response status code | | <input checked="" type="checkbox"/> | 200 |
| 1 Response duration | | <input checked="" type="checkbox"/> | 00:00:00.2150000 |
| 2 Response error type | | <input checked="" type="checkbox"/> | None |
| 3 Response error message | | <input checked="" type="checkbox"/> | |
| 4 Response body | | <input checked="" type="checkbox"/> | {"Items":[{"date":"2024-04-30T00:00+03:00","hour": "00:00","price":2399.9...} |
| 5 Endpoint | | <input checked="" type="checkbox"/> | https://seffaflik.epias.com.tr/electricity-service/v1/markets/dam/data/mcp |

Die abgefragten Daten befinden sich als Wert im Signal *Response Body*, in diesem Fall also ein Textsignal.

Um die einzelnen Werte als Texte oder Zahlen weiterverarbeiten zu können, legen Sie ein Texttrenner-Modul im Bereich *Analytik* des I/O-Managers an. Mit dem Texttrenner-Modul können Sie den großen Datenblock so zerlegen, wie Sie es benötigen.

4.4 Texttrenner-Modul konfigurieren

Fügen Sie ein Texttrenner-Modul hinzu und wählen Sie in den Moduleinstellungen *Allgemein* als Quellsignal das Textsignal mit dem Abfrageergebnis vom HTTP(S)-Modul („Response Body“) aus.

Da die Antwort im json-Format geliefert wurde, stellen Sie noch Trennungsmodus JSON ein.

TextSplitter_HTTPS (52)

| Allgemein | Analog | Diagnose | | | | | | | | | | | | | | | | |
|--|-----------------------|----------|----------------------|----------------------|---------------------------|-----------------------|-----------|------|------|--------------------|-----------|--|-----------|----|-----------|-------|--------------------------------|-------|
| Grundeinstellungen <table border="1"> <tr> <td>Modultyp</td> <td>Texttrenner</td> </tr> <tr> <td>Verriegelt</td> <td>Kein</td> </tr> <tr> <td>Aktiviert</td> <td>True</td> </tr> <tr> <td>Name</td> <td>TextSplitter_HTTPS</td> </tr> <tr> <td>Kommentar</td> <td></td> </tr> <tr> <td>Modul Nr.</td> <td>52</td> </tr> <tr> <td>Zeitbasis</td> <td>10 ms</td> </tr> <tr> <td>Modulname als Präfix verwenden</td> <td>False</td> </tr> </table> | | | Modultyp | Texttrenner | Verriegelt | Kein | Aktiviert | True | Name | TextSplitter_HTTPS | Kommentar | | Modul Nr. | 52 | Zeitbasis | 10 ms | Modulname als Präfix verwenden | False |
| Modultyp | Texttrenner | | | | | | | | | | | | | | | | | |
| Verriegelt | Kein | | | | | | | | | | | | | | | | | |
| Aktiviert | True | | | | | | | | | | | | | | | | | |
| Name | TextSplitter_HTTPS | | | | | | | | | | | | | | | | | |
| Kommentar | | | | | | | | | | | | | | | | | | |
| Modul Nr. | 52 | | | | | | | | | | | | | | | | | |
| Zeitbasis | 10 ms | | | | | | | | | | | | | | | | | |
| Modulname als Präfix verwenden | False | | | | | | | | | | | | | | | | | |
| Quelle <table border="1"> <tr> <td>Quellsignal</td> <td>[51:4] Response body</td> </tr> </table> | | | Quellsignal | [51:4] Response body | | | | | | | | | | | | | | |
| Quellsignal | [51:4] Response body | | | | | | | | | | | | | | | | | |
| Modul Struktur <table border="1"> <tr> <td>Anzahl Analogsignale</td> <td>140</td> </tr> <tr> <td>Anzahl Digitalsignale</td> <td>0</td> </tr> </table> | | | Anzahl Analogsignale | 140 | Anzahl Digitalsignale | 0 | | | | | | | | | | | | |
| Anzahl Analogsignale | 140 | | | | | | | | | | | | | | | | | |
| Anzahl Digitalsignale | 0 | | | | | | | | | | | | | | | | | |
| Verarbeitung <table border="1"> <tr> <td>Trennungsmodus</td> <td>JSON</td> </tr> <tr> <td>Verhalten bei JSON-Teilen</td> <td>Letzte Werte behalten</td> </tr> </table> | | | Trennungsmodus | JSON | Verhalten bei JSON-Teilen | Letzte Werte behalten | | | | | | | | | | | | |
| Trennungsmodus | JSON | | | | | | | | | | | | | | | | | |
| Verhalten bei JSON-Teilen | Letzte Werte behalten | | | | | | | | | | | | | | | | | |
| Name Der Name des Moduls | | | | | | | | | | | | | | | | | | |

Das neue Texttrenner-Modul hat noch die leeren Signaltabellen im Register *Analog*. Wenn aber bereits Daten empfangen wurden, dann wird im Register *Analog* unter der Signaltabelle der empfangene Text angezeigt. Nun brauchen Sie nur auf den Button <Alle JSON-Werte hinzufü-

gen> klicken, und die einzelnen Signale werden aus dem json-Script ausgelesen und als Analogsignale angelegt.

| TextSplitter_HTTPS (52) | | | | | | |
|-------------------------|-------------------|---------|-------------------|----------|--------------------------|-------------------------------------|
| | Allgemein | Analog | Diagnose | | | |
| 0 | items[0].date | Einheit | JSON Pfad | Datentyp | Filter | Aktiv |
| 1 | items[0].hour | | items[0].hour | STRING | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 2 | items[0].price | | items[0].price | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | items[0].priceUsd | | items[0].priceUsd | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | items[0].priceEur | | items[0].priceEur | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | items[1].date | | items[1].date | STRING | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 6 | items[1].hour | | items[1].hour | STRING | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 7 | items[1].price | | items[1].price | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 8 | items[1].priceUsd | | items[1].priceUsd | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 9 | items[1].priceEur | | items[1].priceEur | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 10 | items[2].date | | items[2].date | STRING | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 11 | items[2].hour | | items[2].hour | STRING | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 12 | items[2].price | | items[2].price | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 13 | items[2].priceUsd | | items[2].priceUsd | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | items[2].priceEur | | items[2].priceEur | FLOAT | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Der gelbe Teil hebt den ausgewählten Teil hervor. Um den ausgewählten Teil zu ändern, können Sie die Spalte "JSON Pfad" im Signalraster bearbeiten. Sie können ihn auch ändern, indem Sie auf die gewünschte Zeile in der Vorschau klicken, während die Spalte "JSON Pfad" ausgewählt ist.

[Alle JSON-Werte hinzufügen](#)

```
{
  "items": [
    {
      "date": "2024-04-30T00:00:00+03:00",
      "hour": "00:00",
      "price": 2399.99,
      "priceUsd": 73.95,
      "priceEur": 68.91
    },
    {
      "date": "2024-04-30T01:00:00+03:00",
      "hour": "01:00",
      "price": 2363.14,
      "priceUsd": 72.82,
      "priceEur": 67.86
    },
    {
      "date": "2024-04-30T02:00:00+03:00",
      "hour": "02:00",
      "price": 1631.98,
      "priceUsd": 50.29,
      "priceEur": 46.86
    }
  ]
}
```

Nun stehen die Werte für die weitere Verarbeitung zur Verfügung, z. B. um sie mit *ibaQPanel* anzuzeigen:

| Prices | | | | |
|------------|-------|---------|-----------|-----------|
| Date | Hour | Price | Price USD | Price EUR |
| 2024-04-30 | 00:00 | 2399,99 | 73,95 | 68,91 |
| 2024-04-30 | 01:00 | 2363,14 | 72,82 | 67,86 |
| 2024-04-30 | 02:00 | 1631,98 | 50,29 | 46,86 |
| 2024-04-30 | 03:00 | 1631,98 | 50,29 | 46,86 |
| 2024-04-30 | 04:00 | 1685,00 | 50,29 | 46,86 |
| 2024-04-30 | 05:00 | 1685,00 | 51,92 | 48,38 |
| 2024-04-30 | 06:00 | 999,99 | 30,81 | 28,71 |
| 2024-04-30 | 07:00 | 1631,97 | 50,29 | 46,86 |

5 Diagnose

5.1 Lizenz

Falls die gewünschte Schnittstelle nicht im Signalbaum angezeigt wird, können Sie entweder in *ibaPDA* im I/O-Manager unter *Allgemein – Einstellungen* oder in der *ibaPDA* Dienststatus-Applikation überprüfen, ob Ihre Lizenz für diese Schnittstelle ordnungsgemäß erkannt wird. Die Anzahl der lizenzierten Verbindungen ist in Klammern angegeben.

Die folgende Abbildung zeigt beispielhaft die Lizenz für die Schnittstelle *Codesys-Xplorer*.



5.2 Protokolldateien

Wenn Verbindungen zu Zielsystemen bzw. Clients hergestellt wurden, dann werden alle verbindungsspezifischen Aktionen in einer Textdatei protokolliert. Diese (aktuelle) Datei können Sie z. B. nach Hinweisen auf mögliche Verbindungsprobleme durchsuchen.

Die Protokolldatei können Sie über den Button <Protokolldatei öffnen> öffnen. Der Button befindet sich im I/O-Manager:

- bei vielen Schnittstellen in der jeweiligen Schnittstellenübersicht
- bei integrierten Servern (z. B. OPC UA-Server) im Register Diagnose.

Im Dateisystem auf der Festplatte finden Sie die Protokolldateien von *ibaPDA*-Server ([... \ProgramData\iba\ibaPDA\Log](#)). Die Dateinamen der Protokolldateien werden aus der Bezeichnung bzw. Abkürzung der Schnittstellenart gebildet.

Dateien mit Namen **Schnittstelle.txt** sind stets die aktuellen Protokolldateien. Dateien mit Namen **Schnittstelle_yyyy_mm_dd_hh_mm_ss.txt** sind archivierte Protokolldateien.

Beispiele:

- **ethernetipLog.txt** (Protokoll von EtherNet/IP-Verbindungen)
- **AbEthLog.txt** (Protokoll von Allen-Bradley-Ethernet-Verbindungen)
- **OpcUASeverLog.txt** (Protokoll von OPC UA-Server-Verbindungen)

5.3 Fehlermeldungen

Während der Validierung können folgende Warnungen oder Fehlermeldungen erscheinen:

| Warn-/Fehlermeldung | Mögliche Ursache bzw. Abhilfe |
|---|---|
| Der Endpunkt kann nicht leer sein | Das Feld <i>Endpunkt</i> muss einen Wert enthalten. |
| Der Endpunkt muss mit http, https oder einem Platzhalter beginnen | Das Feld <i>Endpunkt</i> hat einen Eintrag, aber dieser beginnt nicht mit <i>http</i> , <i>https</i> oder einem Platzhalter. Stellen Sie der Adresse <i>http</i> oder <i>https</i> voran, oder einen Platzhalter, der während der Erfassung mit <i>http</i> oder <i>https</i> ersetzt wird. Beispiele: <i>http://iba-ag.com</i> <i>https://iba-ag.com</i> <i>{protocol}://iba-ag.com</i> |
| Dem Platzhalter [...] ist kein Signal zugewiesen | Es wird ein Platzhalter verwendet, aber es ist kein Signal zugewiesen. |

5.4 Verbindungsdiagnose mittels PING

Ping ist ein System-Befehl, mit dem Sie überprüfen können, ob ein bestimmter Kommunikationspartner in einem IP-Netzwerk erreichbar ist.

1. Öffnen Sie eine Windows Eingabeaufforderung.



2. Geben Sie den Befehl "ping" gefolgt von der IP-Adresse des Kommunikationspartners ein und drücken Sie <ENTER>.

→ Bei bestehender Verbindung erhalten Sie mehrere Antworten.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Bytes=32 Zeit=1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
              (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

→ Bei nicht bestehender Verbindung erhalten Sie Fehlermeldungen.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Zielhost nicht erreichbar.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 1, Verloren = 3
              (75% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

5.5 Diagnosemodule

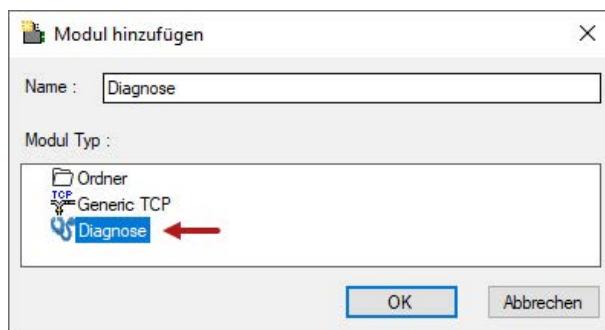
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfasst werden.

Ein Diagnosemodul ist stets einem Datenerfassungsmodul derselben Schnittstelle zugeordnet und stellt dessen Verbindungsinformationen zur Verfügung. Durch die Nutzung eines Diagnosemoduls können die Diagnoseinformationen auch im *ibaPDA*-System durchgängig aufgezeichnet und ausgewertet werden. Diagnosemodule verbrauchen keine Verbindung der Lizenz, da sie keine Verbindung aufbauen, sondern auf ein anderes Modul verweisen.

Nutzungsbeispiele für Diagnosemodule:

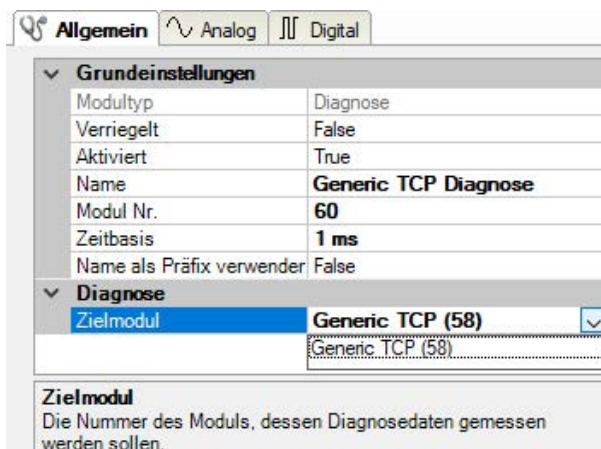
- Wenn der Fehlerzähler einer Kommunikationsverbindung einen bestimmten Wert überschreitet oder eine Verbindung abbricht, kann eine Benachrichtigung generiert werden.
- Bei einem Störungsfall können die aktuellen Antwortzeiten im Telegrammverkehr in einem Störungsreport dokumentiert werden.
- Der Status der Verbindungen kann in *ibaQPanel* visualisiert werden.
- Diagnoseinformationen können über den in *ibaPDA* integrierten SNMP-Server oder OPC DA/UA-Server an übergeordnete Überwachungssysteme wie Netzwerkmanagement-Tools weitergegeben werden.

Wenn für eine Schnittstelle ein Diagnosemodul verfügbar ist, wird im Dialog "Modul hinzufügen" der Modultyp "Diagnose" angezeigt (Beispiel: Generic TCP).



Moduleinstellungen Diagnosemodul

Bei einem Diagnosemodul können Sie folgende Einstellungen vornehmen (Beispiel: Generic TCP):



Die Grundeinstellungen eines Diagnosemoduls entsprechen denen der anderen Module. Es gibt nur eine für das Diagnosemodul spezifische Einstellung, die vorgenommen werden muss: das Zielmodul.

Mit der Auswahl des Zielmoduls weisen Sie das Diagnosemodul dem Modul zu, dessen Verbindungsinformationen erfasst werden sollen. In der Auswahlliste der Einstellung stehen die unterstützten Module derselben Schnittstelle zur Auswahl. Pro Diagnosemodul kann genau ein Dateinfassungsmodul zugeordnet werden. Wenn Sie ein Modul ausgewählt haben, werden in den Registern *Analog* und *Digital* umgehend die verfügbaren Diagnosesignale hinzugefügt. Welche Signale das sind, hängt vom Schnittstellentyp ab. Im nachfolgenden Beispiel sind die Analogwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

| Name | Einheit | Gain | Offset | Aktiv | Istwert |
|---------------------------------|---------|------|--------|-------------------------------------|---------|
| 0 IP-Adresse (Teil 1) | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 1 IP-Adresse (Teil 2) | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 2 IP-Adresse (Teil 3) | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 3 IP-Adresse (Teil 4) | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 4 Port | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 5 Telegrammzähler | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 6 Unvollständig | | 1 | 0 | <input checked="" type="checkbox"/> | |
| 7 Paketgröße (aktuell) | Bytes | 1 | 0 | <input checked="" type="checkbox"/> | |
| 8 Paketgröße (max) | Bytes | 1 | 0 | <input checked="" type="checkbox"/> | |
| 9 Zeit zwischen Daten (aktuell) | ms | 1 | 0 | <input checked="" type="checkbox"/> | |
| 10 Zeit zwischen Daten (min) | ms | 1 | 0 | <input checked="" type="checkbox"/> | |
| 11 Zeit zwischen Daten (max) | ms | 1 | 0 | <input checked="" type="checkbox"/> | |

Die IP(v4)-Adresse eines Generic TCP-Moduls, z. B. (siehe Abbildung), wird entsprechend der 4 Bytes bzw. Oktette in 4 Teile zerlegt, um sie leichter lesen und vergleichen zu können. Andere Größen, wie Portnummer, Zählerstände für Telegramme und Fehler, Datengrößen und Laufzeiten für Telegramme werden ebenfalls ermittelt. Im nachfolgenden Beispiel sind die Digitalwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

| Name | Aktiv | Istwert |
|----------------------------|-------------------------------------|---------|
| 0 Aktiver Verbindungsmodus | <input checked="" type="checkbox"/> | |
| 1 Ungültiges Paket | <input checked="" type="checkbox"/> | |
| 2 Verbinde | <input checked="" type="checkbox"/> | |
| 3 Verbunden | <input checked="" type="checkbox"/> | |

Diagnosesignale

Abhängig vom Schnittstellentyp stehen folgende Signale zur Verfügung:

| Signalname | Bedeutung |
|---|--|
| Aktiv | Nur für redundante Verbindungen relevant. Aktiv bedeutet, dass die Verbindung zur Messung der Daten verwendet wird, d. h. bei redundanten Standby-Verbindungen steht der Wert 0. Bei normalen/nicht redundanten Verbindungen steht immer der Wert 1. |
| Aktualisierungszeit (Istwert/konfiguriert/max/min/Mittelwert) | Gibt die Aktualisierungszeit an, in der die Daten aus der SPS, der CPU oder vom Server abgerufen werden sollen (konfiguriert). Standard ist gleich dem Parameter "Zeitbasis". Während der Messung kann die reale aktuelle Aktualisierungszeit (Istwert) höher sein als der eingestellte Wert, wenn die SPS mehr Zeit zur Übertragung der Daten benötigt. Wie schnell die Daten wirklich aktualisiert werden, können Sie in der Verbindungstabelle überprüfen. Die minimal erreichbare Aktualisierungszeit wird von der Anzahl der Signale beeinflusst. Je mehr Signale erfasst werden, desto größer wird die Aktualisierungszeit. Max/min/Mittelwert: statische Werte der Aktualisierungszeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler |
| Anforderungen Sendewiederholung | Anzahl der nochmals angeforderten Datentelegramme (in) bei Verlust oder Verspätung |
| Antwortzeit (aktuell/konfiguriert/max/min/Mittelwert) | Antwortzeit ist die Zeit zwischen Messwertanforderung von <i>ibaPDA</i> und Antwort von der SPS bzw. Empfang der Daten. Aktuell: Istwert Max/min/Mittelwert: statische Werte der Antwortzeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler |
| Anzahl Anforderungsbefehle | Zähler für Anforderungstelegramme von <i>ibaPDA</i> an die SPS/CPU |
| Aufgebaute Verbindungen (in) | Anzahl der aktuell gültigen Datenverbindungen für den Empfang |
| Aufgebaute Verbindungen (out) | Anzahl der aktuell gültigen Datenverbindungen für das Senden |
| Ausgangsdatenlänge | Länge der Datentelegramme mit Ausgangssignalen in Bytes (<i>ibaPDA</i> sendet) |
| Datenlänge | Länge der Datentelegramme in Bytes |

| Signalname | Bedeutung |
|---|---|
| Datenlänge des Inputs | Länge der Datentelegramme mit Eingangssignalen in Bytes (<i>ibaPDA</i> empfängt) |
| Datenlänge O->T | Größe des Output-Telegramms in Byte |
| Datenlänge T->O | Größe des Input-Telegramms in Byte |
| Definierte Topics | Anzahl der definierten Topics |
| Empfangene Telegramme seit Konfiguration | Anzahl der empfangenen Datentelegramme (in) seit Beginn der Erfassung |
| Empfangene Telegramme seit Verbindungsstart | Anzahl der empfangenen Datentelegramme (in) seit Beginn des letzten Verbindungsaufbaus |
| Empfangszähler | Anzahl der empfangenen Telegramme |
| Exchange ID | ID des Datenaustauschs |
| Falscher Telegrammtyp | Anzahl der Empfangstelegramme mit falschem Telegrammtyp |
| Fehlerzähler | Zähler der Kommunikationsfehler |
| Gepufferte Anweisungen | Anzahl der noch nicht ausgeführten Anweisungen im Zwischenspeicher |
| Gepufferte Anweisungen sind verloren | Anzahl der gepufferten aber nicht ausgeführten und verlorenen Anweisungen |
| Gesendete Telegramme seit Konfiguration | Anzahl der gesendeten Datentelegramme (out) seit Beginn der Erfassung |
| Gesendete Telegramme seit Verbindungsstart | Anzahl der gesendeten Datentelegramme (out) seit Beginn des letzten Verbindungsaufbaus |
| ID der Verbindung O->T | ID der Verbindung für Output-Daten (vom Zielsystem an <i>ibaPDA</i>) Entspricht der Assembly-Instanznummer |
| ID der Verbindung T->O | ID der Verbindung für Input-Daten (von <i>ibaPDA</i> an Zielsystem) Entspricht der Assembly-Instanznummer |
| IP-Adresse (Teil 1-4) | 4 Oktets der IP-Adresse des Zielsystems |
| IP-Quelladresse (Teil 1-4) O->T | 4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i>) |
| IP-Quelladresse (Teil 1-4) T->O | 4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem) |
| IP-Zieladresse (Teil 1-4) O->T | 4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i>) |
| IP-Zieladresse (Teil 1-4) T->O | 4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem) |
| KeepAlive-Zähler | Anzahl der vom OPC UA-Server empfangenen KeepAlive-Telegramme |
| Lesezähler | Anzahl der Lesezugriffe/Datenanforderungen |
| Multicast Anmeldefehler | Anzahl der Fehler bei Multicast-Anmeldung |
| Nachrichtenzähler | Anzahl der empfangenen Telegramme |
| Paketgröße (aktuell) | Größe der aktuell empfangenen Telegramme |

| Signalname | Bedeutung |
|--|--|
| Paketgröße (max) | Größe des größten empfangenen Telegramms |
| Ping-Zeit (Istwert) | Antwortzeit für ein Ping-Telegramm |
| Port | Portnummer für die Kommunikation |
| Producer ID (Teil 1-4) | Producer-ID als 4 Byte unsigned Integer |
| Profilzähler | Anzahl der vollständig erfassten Profile |
| Pufferdateigröße (aktuell/mittl./max) | Größe der Pufferdatei zum Zwischenspeichern der Anweisungen |
| Pufferspeichergröße (aktuell/mittl./max) | Größe des belegten Arbeitsspeichers zum Zwischen-speichern der Anweisungen |
| Schreibverlustzähler | Anzahl missglückter Schreibzugriffe |
| Schreibzähler | Anzahl erfolgreicher Schreibzugriffe |
| Sendezähler | Anzahl der Sendetelegramme |
| Sequenzfehler | Anzahl Sequenzfehler |
| Synchronisation | Gerät wird für die isochrone Erfassung synchronisiert |
| Telegramme pro Zyklus | Anzahl der Telegramme im Zyklus der Aktualisierungszeit |
| Telegrammzähler | Anzahl der empfangenen Telegramme |
| Topics aktualisiert | Anzahl der aktualisierten Topics |
| Trennungen (in) | Anzahl der aktuell unterbrochenen Datenverbindungen für den Empfang |
| Trennungen (out) | Anzahl der aktuell unterbrochenen Datenverbindungen für das Senden |
| Unbekannter Sensor | Anzahl unbekannter Sensoren |
| Ungültiges Paket | Ungültiges Datenpaket erkannt |
| Unvollständig | Anzahl unvollständiger Telegramme |
| Unvollständige Fehler | Anzahl unvollständiger Telegramme |
| Verarbeitete Anweisungen | Anzahl der ausgeführten SQL-Anweisungen seit dem letzten Start der Erfassung |
| Verbinde | Verbindung wird aufgebaut |
| Verbindungsphase (in) | Zustand der ibaNet-E Datenverbindung für den Empfang |
| Verbindungsphase (out) | Zustand der ibaNet-E Datenverbindung für das Senden |
| Verbindungsversuche (in) | Anzahl der Versuche, die Empfangsverbindung (in) aufzubauen |
| Verbindungsversuche (out) | Anzahl der Versuche, die Sendeverbindung (out) aufzubauen |
| Verbunden | Verbindung ist aufgebaut |
| Verbunden (in) | Eine gültige Datenverbindung für den Empfang (in) ist vorhanden |
| Verbunden (out) | Eine gültige Datenverbindung für das Senden (out) ist vorhanden |

| Signalname | Bedeutung |
|--|---|
| Verlorene Images | Anzahl der verlorenen Images (in), die selbst nach einer Sendewiederholung nicht empfangen wurden |
| Verlorene Profile | Anzahl unvollständiger/fehlerhafter Profile |
| Zeilen (letzte) | Anzahl der Ergebnissezeilen der letzten SQL-Abfrage (innerhalb der projektierten Anzahl von Ergebnissezeilen) |
| Zeilen (Maximum) | Höchste Anzahl der Ergebnissezeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projektierten Anzahl von Ergebnissezeilen) |
| Zeit zwischen Daten (aktuell/max/min) | Zeit zwischen zwei korrekt empfangenen Telegrammen Aktuell: zwischen den letzten zwei Telegrammen Max/min: statistische Werte seit Start der Erfassung oder Rücksetzen der Zähler |
| Zeit-Offset (Istwert) | Gemessene Zeitdifferenz der Synchronität zwischen dem ibaNet-E-Gerät und <i>ibaPDA</i> |

6 Support und Kontakt

Support

Tel.: +49 911 97282-14

E-Mail: support@iba-ag.com

Hinweis



Wenn Sie Support benötigen, dann geben Sie bitte bei Softwareprodukten die Nummer des Lizenzcontainers an. Bei Hardwareprodukten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

Kontakt

Hausanschrift

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland

Tel.: +49 911 97282-0

E-Mail: iba@iba-ag.com

Postanschrift

iba AG
Postfach 1828
90708 Fürth

Warenanlieferung, Retouren

iba AG
Gebhardtstraße 10
90762 Fürth

Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite:

www.iba-ag.com