



# Affiliate SOAP

## **Inhalt:**

SOAP-Statistik - Schnittstellenbeschreibung

- URI
- Methode
- Authentifizierung
- Detaillierte Beschreibung des Soap-Service
- Suchen von Transaktionsdaten
- Statistiken anfordern
- Rückgabewerte der Soap-Methoden
- Beispiele

SOAP-Transaktionsbearbeitung - Clientseitige Programmierbeispiele

- perl
- php

Web Service Definition Datei iqx\_downlink.wsdl

# SOAP-Statistik Schnittstellenbeschreibung

## URI

### SOAP URI:

[https:// www.qc-domain.de/iqx\\_downlink](https://www.qc-domain.de/iqx_downlink)

### SOAP PROXY:

[https://www.qc-domain.de/iqx\\_downlink\\_soap.cgi](https://www.qc-domain.de/iqx_downlink_soap.cgi)

## Methode

Sie senden mittels SOAP-Request (z.B. über DotNET/SOAP, Java, PHP) eine Anfrage an die folgende URL auf unserem Server:

**[https://www.qc-domain.de/iqx\\_downlink\\_soap.cgi](https://www.qc-domain.de/iqx_downlink_soap.cgi)**

Die angeforderten Daten werden im XML-Format als SOAP-Response geliefert.

Eine wsdl-Datei mit der formalen Beschreibung des Soap-Services, der Methoden und Parameter liegt bereit unter folgender URL:

**[https://www.qc-domain.de/scripts/soap\\_doc/iqx\\_downlink\\_soap.wsdl.cgi](https://www.qc-domain.de/scripts/soap_doc/iqx_downlink_soap.wsdl.cgi)**

## Authentifizierung

- IDENT-Code (z.B. "QCX"), PartnerID und Passphrase (z.B. "un58jin44ap96uszkx2qq") im XML-Inhalt
- Zugriffs-Beschränkung auf eine IP-Adresse

## Soap-Methoden

### Suchen von Transaktionsdaten

Name der Methode: search

Name	Beschreibung	Format	Beispielwert
ident	Authentifizierungs-ID	Text (3 Zeichen)	
keyphrase	Zugangspasswort	Text (40 Zeichen)	
cpid	Kampagnen-ID (aus QC)	Zahl	2
partner_id	Affiliate / Partner-ID (aus QC)	Zahl	5
status	Ein oder mehrere mit Komma getrennte Status open, cleared, storno, paid	Text	paid, cleared
order_date	Bestelldatum Zeitraum: von - bis	DD.MM.YYYY-DD.MM.YYYY	10.08.2015-12.08.2015
delivery_date	Lieferdatum Zeitraum: von - bis	DD.MM.YYYY-DD.MM.YYYY	12.08.2015-14.08.2015

**Rot markierte Felder** sind Pflichtfelder.

### Statistik anfordern

Name der Methode: stat

Name	Beschreibung	Format	Beispielwert
ident	Authentifizierungs-ID	Text (3 Zeichen)	
keyphrase	Zugangspasswort	Text (40 Zeichen)	
cpid	Kampagnen-ID (aus QC)	Zahl	2
partner_id	Affiliate / Partner-ID (aus QC)	Zahl	5
sub_id	Sub-ID des Partners	Text	2a9ahs626a5
order_date	Bestelldatum Zeitraum: von - bis	DD.MM.YYYY-DD.MM.YYYY	10.08.2015-12.08.2015
delivery_date	Lieferdatum Zeitraum: von - bis	DD.MM.YYYY-DD.MM.YYYY	12.08.2015-14.08.2015

**Rot markierte Felder** sind Pflichtfelder.

## Rückgabewerte der SOAP-Methoden

list_count:	Anzahl Ergebnisse der Suche
list_result:	Array der gefundenen Transaktionsinformationen
response_code:	OK oder ERR
status_message:	siehe unten

### **Beschreibung der Rückgabewerte**

OK	found XXX transactions
ERR	wrong xml parameter or data
ERR	incorrect ident 'ABC'
ERR	keyphrase wrong 'un58jin44ap96uszkx2q?
ERR	ident 'ABC' or keyphrase ' un58jin44ap96uszkx2q' wrong
ERR	incorrect status
ERR	too much results (>50.000). please add more criterias to your query

## Beispiele

### Suchen nach Transaktionen

Zum Beispiel: Suche nach Transaktionen vom 05.05.2010, PartnerID 1, Status beliebig

#### Anfrage:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <search xmlns="http://qc5.netslave.de/iqx_downlink">
    <iqx_downlink>
      <ident>QC5</ident>
      <cpid>1</cpid>
      <keyphrase>fmasaw0607dm0cx734bsyb7482n3fgyxnz8engf7</keyphrase>
      <status></status>
      <partner_id>1</partner_id>
      <order_date>05.05.2010</order_date>
    </iqx_downlink>
  </search>
</soap:Body>
</soap:Envelope>
```

#### Antwort:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<soap:Body>
  <statResponse xmlns="http://qc5.netslave.de/iqx_downlink">
    <iqx_downlink xmlns:ns1="http://qc5.netslave.de/iqx_downlink">
      <list_result>
        <statistic>
          <open_commissions xsi:type="xsd:float">0.659999996423721</open_commissions>
          <total_transactions xsi:type="xsd:int">3</total_transactions>
          <epc xsi:type="xsd:float">0.659999996423721</epc>
          <confirmed_commissions xsi:type="xsd:int">0</confirmed_commissions>
          <date xsi:type="xsd:date">2010-05-05</date>
          <ecpa xsi:type="xsd:int">0</ecpa>
          <open_transactions xsi:type="xsd:int">3</open_transactions>
          <klicks xsi:type="xsd:int">0</klicks>
          <views xsi:type="xsd:int">0</views>
          <cr xsi:type="xsd:float">0.0</cr>
          <turnover xsi:type="xsd:int">3</turnover>
          <confirmed_transactions xsi:type="xsd:int">0</confirmed_transactions>
          <commission xsi:type="xsd:float">0.659999996423721</commission>
        </statistic>
      </list_result>
    </iqx_downlink>
  </statResponse>
</soap:Body>
</soap:Envelope>
```

```

</list_result>
<status_message xsi:type="xsd:string">found 1 transactions</status_message>
<list_count xsi:type="xsd:string">1</list_count>
<response_code xsi:type="xsd:string">OK</response_code>
</iqx_downlink>
</statResponse>
</soap:Body>
</soap:Envelope>

```

## Statistik anfordern

Zum Beispiel: Statistik zu Kampagne 3, Partner 1, Zeitraum vom 5.5.2010 bis 6.5.2010

### Anfrage:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <search xmlns="http://qc5.netslave.de/iqx_downlink">
    <iqx_downlink>
      <ident>QC5</ident>
      <cpid>3</cpid>
      <delivery_date>05.05.2010-06.05.2010</delivery_date>
      <keyphrase>fmasaw0607dm0cx734bsyb7482n3fgyxnz8engf7</keyphrase>
      <partner_id>1</partner_id>
    </iqx_downlink>
  </search>
</soap:Body>
</soap:Envelope>

```

### Antwort:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<soap:Body>
<statResponse xmlns="http://qc5.netslave.de/iqx_downlink">
<iqx_downlink xmlns:ns1="http://qc5.netslave.de/iqx_downlink">
<list_result>
  <statistic>
    <open_commissions xsi:type="xsd:float">0.659999996423721</open_commissions>
    <total_transactions xsi:type="xsd:int">3</total_transactions>
    <epc xsi:type="xsd:float">0.659999996423721</epc>
    <confirmed_commissions xsi:type="xsd:int">0</confirmed_commissions>
    <date xsi:type="xsd:date">2010-05-05</date>
    <ecpa xsi:type="xsd:int">0</ecpa>
    <open_transactions xsi:type="xsd:int">3</open_transactions>

```

```
<klicks xsi:type="xsd:int">0</klicks>
<views xsi:type="xsd:int">0</views>
<cr xsi:type="xsd:float">0.0</cr>
<turnover xsi:type="xsd:int">3</turnover>
<confirmed_transactions xsi:type="xsd:int">0</confirmed_transactions>
<commission xsi:type="xsd:float">0.659999996423721</commission>
</statistic>
</list_result>
<status_message xsi:type="xsd:string">found 1 transactions</status_message>
<list_count xsi:type="xsd:string">1</list_count>
<response_code xsi:type="xsd:string">OK</response_code>
</iqx_downlink>
</statResponse>
</soap:Body>
</soap:Envelope>
```

# SOAP-Transaktionsbearbeitung - Clientseitige Programmierbeispiele

perl

Beispiel Clientseitige Programmierung in perl mit Modul SOAP::Lite

```
#!/usr/bin/perl
-w use
SOAP::Lite ; use
Encode;
use Data::Dumper;

my $soap_uri      = 'https://www.qc-domain.de/iqx_downlink';
my $soap_proxy    = 'https://www.qc-domain.de/iqx_downlink_soap.cgi';

my $soap = SOAP::Lite-> uri($soap_uri)-> proxy($soap_proxy)-> encoding('UTF-8'); my
$string = 'Gartenmöbel';
my $ustring= $string;
Encode::from_to($ustring, "iso-8859-1", "utf8");    ## anstelle von use utf8;

my $result = $soap->search( SOAP::Data->name('iqx_downlink'
=> \SOAP::Data->value(
    SOAP::Data->name('ident')->type('string' => 'ABC'), SOAP::Data-
>name('keyphrase')->type('string' => 'un58jin44ap9h57h9rml'), SOAP::Data-
>name('partner_id')->type('string' => '1'), SOAP::Data->name('cpid')-
>type('string' => '1'), SOAP::Data->name('order_date')->type('string' =>
"01.01.2010-06.05.2010"),
)) );

unless ($result->fault) {
    my $response = $result->result();
    print $response->{response_code}, ": ", $response->{status_message}, "\n" ; }
else {
    print join ', ', $re-
sult->faultcode,
    $result-
>faultstring, $re-
sult->faultdetail;
}
```

php

Setzt voraus, dass soap Support in php konfiguriert wurde

```
<?php
$client = new SoapClient('https://www.qc-
domain.de/soap_doc/iqx_downlink.wsdl', ar-
ray('encoding'=>'UTF-8' ));
$parameter = array(
    "ident" => "ABC",
    "keyphrase" => "un58jin44ap9h57h9rml",
    "cpid" => "1",
    "partner_id" =>"1",
    "order_date" =>"01.01.2010-06.05.2010",
);
$result = $client->search( $parameter );
echo $result->response_code, ": ", $result-
>status_message, "\n"; ?>
```



# Web Service Definition Datei iqx\_downlink.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:enc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="urn:iqx_downlink"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="iqx_downlink" targetNamespace="urn:iqx_downlink">
  <documentation>Die WSDL Spezifikation fuer iqx_downlink</documentation>
  <types>
    <documentation>Dieser Abschnitt definiert die Typen, die im Folgenden verwendet werden,
um diejenigen Daten zu beschreiben, die die Transaktionen ausmachen. Nur die Datentypen, die
nicht bereits im XML Schema verfügbar sind, werden hier definiert.</documentation>
    <xsd:schema targetNamespace="urn:iqx_downlink">
      <xsd:complexType name="AuthData">
        <xsd:all>
          <xsd:element name="ident" type="xsd:string" minOccurs="1" />
          <xsd:element name="keyphrase" type="xsd:string" minOccurs="1" />
          <xsd:element name="partner_id" type="xsd:string" minOccurs="1" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="TransactionSearch">
        <xsd:all>
          <xsd:element name="ident" type="xsd:string" minOccurs="1" />
          <xsd:element name="keyphrase" type="xsd:string" minOccurs="1" />
          <xsd:element name="cpid" type="xsd:string" minOccurs="1" />
          <xsd:element name="partner_id" type="xsd:string" minOccurs="1" />
          <xsd:element name="order_date" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="TransactionItem">
        <xsd:all>
          <xsd:element name="cpid" type="xsd:string" minOccurs="1" />
          <xsd:element name="sub_id" type="xsd:string" minOccurs="1" />
          <xsd:element name="partner_id" type="xsd:string" minOccurs="1" />
          <xsd:element name="transaction_id" type="xsd:string" minOccurs="1" />
          <xsd:element name="status" type="xsd:string" />
          <xsd:element name="amount" type="xsd:string" />
          <xsd:element name="annex_id" type="xsd:string" />
          <xsd:element name="product_id" type="xsd:string" />
          <xsd:element name="product" type="xsd:string" />
          <xsd:element name="order_date" type="xsd:string" />
          <xsd:element name="delivery_date" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="ArrayOfTransactionItem">
        <xsd:complexContent>
          <xsd:restriction base="enc:Array">
            <xsd:attribute ref="enc:arrayType" wsdl:arrayType="tns:TransactionItem[]" />
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="TransactionResponse">
        <xsd:all>
          <xsd:element name="response_code" type="xsd:string" minOccurs="1" />
          <xsd:element name="status_message" type="xsd:string" minOccurs="1" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="TransactionResponseList">
        <xsd:all>
          <xsd:element name="response_code" type="xsd:string" minOccurs="1" />
          <xsd:element name="status_message" type="xsd:string" minOccurs="1" />
          <xsd:element name="list_count" type="xsd:string" />
          <xsd:element name="list_result" type="tns:ArrayOfTransactionItem" />
        </xsd:all>
      </xsd:complexType>
    </xsd:schema>
  </types>

```

```

<message name="iqxSearchRequest">
  <documentation>Nachrichtenformat zu Auflisten von Transaktionen</documentation>
  <part name="iqx_downlink" type="tns:TransactionSearch" />
</message>
<message name="iqxResponse">
  <documentation>Nachrichtenformat zu Antwort auf Aenderung von Transaktio-
nen</documentation>
  <part name="iqx_downlink" type="tns:TransactionResponse" />
</message>
<message name="iqxResponseList">
  <documentation>Nachrichtenformat zu Antwort auf Suchen/Auflisten von Transaktio-
nen</documentation>
  <part name="iqx_downlink" type="tns:TransactionResponseList" />
</message>
<portType name="iqxPort">
  <documentation>Dieser port stellt alle Operationen von iqx_downlink zur Verfue-
gung</documentation>
  <operation name="search">
    <documentation>Auflistung von Transaktionen

Suche nach Transaktionen mit gegebenen Suchkriterien</documentation>
    <input message="tns:iqxSearchRequest" />
    <output message="tns:iqxResponseList" />
  </operation>
  <operation name="stat">
    <documentation>Auflistung von Statistiken

Statistiken zu Transaktionen mit gegebenen Suchkriterien</documentation>
    <input message="tns:iqxSearchRequest" />
    <output message="tns:iqxResponseList" />
  </operation>
</portType>
<binding name="iqxBinding" type="tns:iqxPort">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="search">
    <soap:operation soapAction="urn:iqx_downlink#search" />
    <input>
      <soap:body use="encoded" namespace="urn:iqx_downlink" encod-
ingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:iqx_downlink" encod-
ingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="stat">
    <soap:operation soapAction="urn:iqx_downlink#search" />
    <input>
      <soap:body use="encoded" namespace="urn:iqx_downlink" encod-
ingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:iqx_downlink" encod-
ingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>
<service name="iqx_downlink">
  <documentation>Der iqx service definiert die Kombinationen

von portTyp zu Binding die benoetigt werden, um die Funktionalitaet der iqx Klasse allgemein
verfuegbar zu machen</documentation>
  <port name="iqxPort" binding="tns:iqxBinding">
    <documentation>Das Binding von portTyp zu SOAP,

die Adresse verwendet HTTP und verweist auf einen Soap-Server mit der iqx Schnittstel-
le.</documentation>
    <soap:address location="http://qc5.netslave.de/scripts/iqx_downlink_soap.wsdl.cgi" />
  </port>
</service>
</definitions>

```