

API documentation

Inkassogram XML Integration Invoice Bookkeeping

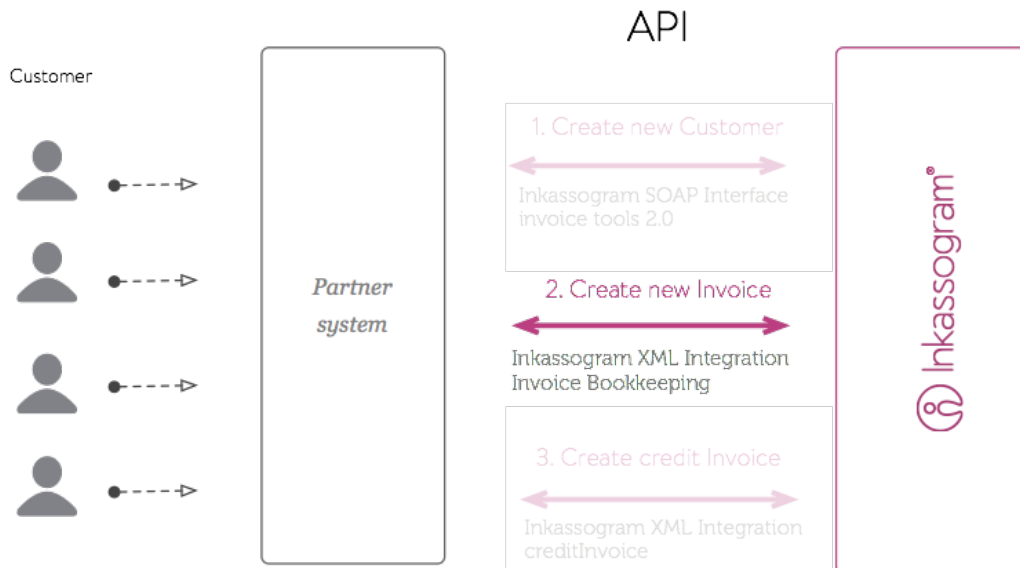
Use this API to create a new debit invoice.

V1.0

Table of contents

| | | |
|----------|-------------------------------------|-----------|
| 1 | Introduction..... | 3 |
| 1.1 | Header Data | 3 |
| 1.2 | XML Types | 4 |
| 2 | XML Integration..... | 5 |
| 2.1 | Create Invoice..... | 5 |
| 2.1.1 | Create Invoice Request XML..... | 6 |
| 2.1.2 | Create Invoice Response XML..... | 7 |
| 2.1.3 | Create Invoice Data Dictionary..... | 7 |
| 2.1.4 | Callback event trigger | 13 |
| 2.1.5 | Create Invoice XSD Scheme | 14 |
| 3 | Status Codes | 18 |
| 4 | Example Code..... | 20 |
| 4.1 | PHP Integration | 20 |
| 4.2 | Java Integration | 20 |
| 4.3 | .NET Integration..... | 20 |

1 Introduction



This document describes the Inkassogram XML Integration format. An API is also available to check the payment status of invoices. The requests are posted in XML format to the specified URL using a HTTP/HTTPS post request.

Note that the HTTP Content-Type must be set to “text/xml” and encoding to UTF-8.

This document explains the API, provides a data dictionary detailing the format of content and provides sample calls to the API. Finally some specific notes on the XML format and Unicode are provided.

1.1 Header Data

The following two parameters should be defined in the header for all requests and communication with Inkassogram API.

customerNo: Identify number sent by Inkassogram after signed agreement.

Key: MD5(Public ServerIP, Timestamp, PrivateKey).

Definition of MD5 Key

Public ServerIP: The Public IP address used by the server for the integration API.

Timestamp: Daily date YYYYMMDD (20101224).

PrivateKey: length 32 characters with case sensitive, sent by Inkassogram after signed agreement.

Note: The **Key** is NOT the same as the Private Key. It's an MD5 SUM of the parameters above without colon and spaces.

1.2 XML Types

All fields within the XML API have their type defined in an XML Schema. The data types of the XML elements are typically simple XML Schema types such as strings and integers with restrictions on their length or values. The relevant section for each API includes examples of valid XML as defined by the XML schema. In addition a data dictionary for each API explains the expected format of each XML element, for example see section 2.1.3. The following notation is used to describe the format of the XML content in the data dictionary.

2 XML Integration

Header parameters must be defined for all requests! See section 1.1

Create Invoice Request XSD Schema:

<https://api.inkassogram.se/API/createInvoiceBookkeepingSchema1.0.xsd>

Create Invoice POST Request:

<https://api.inkassogram.se/API/createInvoiceBookkeeping>

2.1 Create Invoice

Use the Create Invoice Bookkeeping API to create a new invoice for the customer. Invoice request are submitted to the Inkassogram platform as XML messages, transmitted as HTTP/HTTPS POST requests. Applications submitting messages to the Create Invoice API must format the XML request as described in section 2.1.1.

Upon receipt of a valid XML request, the Inkassogram platform will prepare a message for delivery and return an XML response as described in section 2.1.2.

When returned unsuccessfully responses, as a time out or such. It's important to make one or two retries before displaying an error for the end user.

To send invoices to foreign customers

If the customer is from the nordic countries their will not be so much to change. Just use the country element in the complex type foreignCustomer. The address lines will be ignored in this case. To change billing address, use the care of element.

If the customer is from UK or somewhere else, you have to define the countryWithNoSsnCheck and specify full contact details of the customer using the address line elements.

Company/Person name, address and all of billing information shall be defined in the foreignCustomer complex type.

Note that customerAddressLine shall not include the country because it's already defined in their own element. And customerAddressLine will be ignored if the element country is used.

The invoice will be sent in available language and if their is no language available english will be default for the foreign customers.

2.1.1 Create Invoice Request XML

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall xmlns="https://api.inkassogram.se/API/createInvoiceBookkeeping"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://api.inkassogram.se/API/createInvoiceBookkeeping
https://api.inkassogram.se/API/createInvoiceBookkeepingSchema1.0.xsd">
  <methodName>createInvoice</methodName>
  <request>
    <testInvoice>true</testInvoice>
    <makeInvoiceReservation>0</makeInvoiceReservation>
    <service>1</service>
    <printSetup>1</printSetup>
    <ssn>5567854616</ssn>
    <careOfAddress>
      <co_name></co_name>
      <co_address></co_address>
      <co_address2></co_address2>
      <co_zip></co_zip>
      <co_city></co_city>
    </careOfAddress>
    <invoiceRef></invoiceRef>
    <shippingFee></shippingFee>
    <expFee></expFee>
    <dueDate>1356912000</dueDate>
    <mobile></mobile>
    <email>techsupport@inkassogram.se</email>
    <orderNo>1</orderNo>
    <ourRef></ourRef>
    <yourRef></yourRef>
    <invoiceRows>
      <row>
        <articleNo></articleNo>
        <text></text>
        <desc></desc>
        <vat></vat>
        <quantity></quantity>
        <price></price>
        <bookkeepingAccount></bookkeepingAccount>
      </row>
      <row>
        <articleNo>4144</articleNo>
        <text>Biljett</text>
        <desc>GOT - STO 2001-01-01</desc>
        <vat>12</vat>
        <quantity>1</quantity>
        <price>3400</price>
        <bookkeepingAccount>3010</bookkeepingAccount>
      </row>
    </invoiceRows>
    <comments></comments>
    <discount></discount>
  </request>
</methodCall>
```

2.1.2 Create Invoice Response XML

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall xmlns="https://api.inkassogram.se/API/createInvoiceBookkeeping"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://api.inkassogram.se/API/createInvoiceBookkeeping
https://api.inkassogram.se/API/createInvoiceBookkeepingSchema1.0.xsd">
  <methodName>createInvoice</methodName>
  <response>
    <statusCode>1</statusCode>
    <ocr></ocr>
    <customerName></customerName>
    <customerAddress></customerAddress>
    <customerZip></customerZip>
    <customerCity></customerCity>
    <companyOrgNo></companyOrgNo>
    <companyName></companyName>
    <companyAddress></companyAddress>
    <companyZip></companyZip>
    <companyCity></companyCity>
    <pdfFile></pdfFile>
    <errorCode></errorCode>
    <bgAccount></bgAccount>
    <pgAccount></pgAccount>
    <ibanAccount></ibanAccount>
    <swiftAccount></swiftAccount>
    <invoiceDisclaimer></invoiceDisclaimer>
    <description></description>
  </response>
</methodCall>
```

2.1.3 Create Invoice Data Dictionary

Explained elements for the XML Request Create Invoice Bookkeeping

| Element | Data Type | Optional | Value | Description |
|------------------------|-----------|----------|-------|--|
| testInvoice | String | Yes | 1 | For testing mode, doesn't save the purchases to DB |
| makeInvoiceReservation | Boolean | Yes | 1/0 | If true the invoice will NOT be sent to the customer until another request is received using SOAP with method activate_invoice |

| Element | Data Type | Optional | Value | Description |
|--------------------|-----------|----------|---|--|
| forceToSend | Boolean | Yes | 1/0 | Force to send invoice even if the end user got bad credit rate. Can be used if the errorCode is 116, 117 120 and 121 |
| service | Integer | No | 1,2,3 | 1 - Invoice Service 2 - Factoring 3 - American Factoring |
| printSetup | Integer | No | 1,2,3,4,5 | 1 - Email, PDF 2 - Terminal print 3 - Print on Demand 4 - SMS Invoice 5 - EDI/SVE Invoice |
| ssn | String | No | YYYYMMDDXXXX / YYMMDDXXXX / XXXXXX-XXXX | Primary Social Security Number OR Organization Number. |
| organizationNumber | String | Yes | XXXXXX-XXXX | Organization Number used when ssn is set to Primary Social Security Number together with BankID. |
| foreignCustomer | | Yes | | Used to send invoices to other countries |
| careOfAddress | | Yes | | To send the invoice to another address instead of the company address. |
| invoiceRef | String | Yes | | This value will be visible on the invoice as an order number. |
| invoiceOrderNo | String | Yes | | The reference number of the order made from the end customer. Will be visible on the invoice for the end customer. |

| Element | Data Type | Optional | Value | Description |
|----------------------|-----------|----------|-------|--|
| languageCode | String | Yes | xx | ISO-639-1 alpha-2 format required. Example: "en" generates an invoice in English. |
| currencyCode | String | Yes | XXX | ISO 4217 format required. Example: "USD" or "DKK". |
| currencyExchangeRate | String | Yes | | Exchange Rate where base currency is currencyCode, and counter currency is SEK. Example: "8.81" for USDSEK. |
| termsOfDelivery | String | Yes | | Terms of delivery |
| cashRounding | Boolean | Yes | 1/0 | If cash rounding should be applied to invoice. |
| vatRoundingPerRow | Boolean | Yes | 1/0 | If ERP rounds incl vat at subtotal level, to calculate total amount of invoice. Default is false. Important: Cannot be defined as false if vatIncluded is set to true. |
| vatIncluded | Boolean | Yes | 1/0 | If invoice should be presented incl VAT. Default is false for organisations, and true for individuals. |
| shippingFee | Integer | Yes | | Shipping fee for an order. This value will be added as an invoice row |
| expFee | Integer | Yes | | Expedition fee. This value will be added as an invoice row |

| Element | Data Type | Optional | Value | Description |
|--------------------|-----------|----------|--|---|
| invoiceDate | Integer | Yes | Unixtime | Date when the invoice shall be sent to the receiver. |
| dueDate | Integer | Yes | Unixtime | Invoice Due Date in Unixtime format |
| callback | String | Yes | URL encoded e.g. http%3A%2F%2FcusterServer.com%2Fcallback%3FinvoiceId%3Dxx1000%26type%3D1 | URL for callback trigger when invoice has been changed or a payment is received. |
| mobile | String | No | | Mobile Phone Number |
| email | String | No | | Email Address |
| orderNo | String | No | | Order Number, must be unique for each invoice and should be an internal value. Used to get the status or credit the invoice. Invoice number from ERP. |
| ourRef | String | Yes | | Visible as Our Reference |
| yourRef | String | Yes | | Visible as Your Reference |
| InvoiceRows | | No | | |
| comments | String | Yes | | Add an invoice Comment for the end user. Visible below the article rows on the invoice. |
| discount | integer | Yes | | Discount percentage of the total invoice amount. |
| billingVar | String | Yes | | Element available for bookkeeping. Data will be visible in sales note |

| Element | Data Type | Optional | Value | Description |
|------------------------------|--------------|----------|--------------------------------|--|
| | | | | files |
| attachedDocument | String | Yes | base 64 encoded binary file | Attached document in base64 encoded format |
| attachedDocumentMd5 | String | Yes | | md5 hash of binary document |
| foreignCustomer | | Yes | | |
| country | Enum, String | Yes | Sweden/Norway/Denmark/Finland | Just countries that we support with ssn_check |
| countryWithNoSsnCheck | | Yes | | |
| countryCode | String | No | SE = Sverige, NO = Norway etc. | Special agreement is needed for use of this |
| addressLine1 | String | No | | Must be used with the correct address if countryWithNoSsnCheck is used. To change address if country is used you have to define the careOfAddress instead. These elements will be ignored if just country is used. |
| addressLine2 | String | No | | Must be used with the correct address if countryWithNoSsnCheck is used |
| addressLine3 | String | Yes | | Extra address line when countryWithNoSsnCheck is used |
| addressLine4 | String | Yes | | Extra address line when countryWithNoSsnCheck is used |
| careOfAddress | | Yes | | |

| Element | Data Type | Optional | Value | Description |
|---------------------|-----------|----------|------------|---|
| co_name | String | No | | Company Name / Invoice Receiver Name |
| co_address | String | No | | Address |
| co_address2 | String | Yes | | Address 2 |
| co_zip | String | Yes | | Zip Code |
| co_city | String | Yes | | City / Town |
| InvoiceRows | | No | | |
| articleNo | String | Yes | | Article Number |
| text | String | No | | Article Text, such as Ticket, Member fee.. Max length 120 characters |
| desc | String | Yes | | Description of the article. Max length 120 characters |
| vat | Integer | Yes | default 25 | Vat of the article |
| quantity | Float | Yes | default 1 | The quantity of the sold articles |
| price | Integer | No | | Price inc. VAT in Å price |
| unit | String | Yes | | Column Unit will be added to the article rows/invoice rows |
| discount | Float | Yes | | Discount percentage of the total invoice amount. |
| bookkeeping account | String | Yes | | Bookkeeping account for the bookkeeping |
| profitUnit | String | Yes | | Used in the bookkeeping |
| project | String | Yes | | Used in the bookkeeping |

2.1.4 Callback event trigger

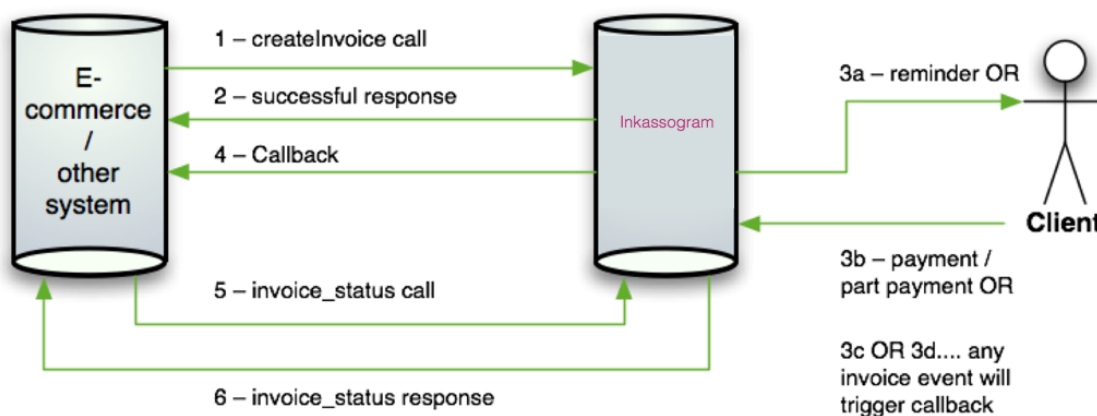
1. The invoice is created using createInvoice method
2. Succeeded response
3. Invoice event appear, such as reminder sent to customer, payment or whatever it is
4. Inkassogram will trigger the Callback URL and receive HTTP code 200 if succeeded, if not 3 retries will be done.
5. Customer handles the callback and can now trigger the invoice_status or invoice_details call using the SOAP interface.
6. response from WS call

Inkassogram will add the option "type" in the GET request.

type = 1: The invoice value/amount left have been changed by a payment reminder or such.

type = 2: The invoice information has been changed.

Example url: <http://customerServer.com/callback?invoiceId=xx1000&type=1>



2.1.5 Create Invoice XSD Schema

URL: <https://api.inkassogram.se/API/createInvoiceBookkeepingSchema1.0.xsd>

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cre="https://api.inkassogram.se/API/createInvoiceBookkeeping"
  targetNamespace="https://api.inkassogram.se/API/createInvoiceBookkeeping"
  elementFormDefault="qualified">
  <xs:element name="methodCall" type="cre:methodCallType" />
  <xs:complexType name="rowType">
    <xs:sequence>
      <xs:element type="xs:string" name="articleNo" maxOccurs="1"
        minOccurs="0" />
      <xs:element type="xs:string" name="text" maxOccurs="1" minOccurs="1" />
      <xs:element type="xs:string" name="desc" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:int" name="vat" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:float" name="quantity" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:int" name="price" maxOccurs="1" minOccurs="1" />
      <xs:element type="xs:string" name="unit" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:int" name="discount" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:int" name="bookkeepingAccount" maxOccurs="1"
        minOccurs="0" />
      <xs:element type="xs:string" name="profitUnit" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:string" name="project" maxOccurs="1" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="careOfType">
    <xs:sequence>
      <xs:element type="xs:string" name="co_name" maxOccurs="1" minOccurs="1" />
      <xs:element type="xs:string" name="co_address" maxOccurs="1" minOccurs="1" />
      <xs:element type="xs:string" name="co_address2" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:string" name="co_zip" maxOccurs="1" minOccurs="0" />
      <xs:element type="xs:string" name="co_city" maxOccurs="1" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="methodCallType">
    <xs:sequence>
      <xs:element type="xs:string" name="methodName" />
      <xs:choice>
        <xs:element type="cre:requestType" name="request"
          maxOccurs="1" minOccurs="0" />
        <xs:element type="cre:responseType" name="response"
          maxOccurs="1" minOccurs="0" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="statusCode" maxOccurs="1" minOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:enumeration value="1" />
            <xs:enumeration value="0" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="xs:string" name="ocr" maxOccurs="1" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        <xs:element type="xs:string" name="customerName" maxOccurs="1" minOccurs="0"
/>
        <xs:element type="xs:string" name="customerAddress" maxOccurs="1"
minOccurs="0" />
        <xs:element type="xs:string" name="customerZip" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="customerCity" maxOccurs="1" minOccurs="0"
/>
        <xs:element type="xs:string" name="pdfFile" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="bgAccount" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="pgAccount" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="ibanAccount" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="swiftAccount" maxOccurs="1" minOccurs="0"
/>
        <xs:element type="xs:string" name="invoiceDisclaimer" maxOccurs="1"
minOccurs="0" />
        <xs:element type="xs:string" name="errorCode" maxOccurs="1" minOccurs="0" />
        <xs:element type="xs:string" name="description" maxOccurs="1" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="invoiceRowsType">
    <xs:sequence>
        <xs:element type="cre:rowType" name="row" maxOccurs="unbounded" minOccurs="1"
/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="careOfAddressType">
    <xs:sequence>
        <xs:element type="cre:careOfType" name="careOf" maxOccurs="1" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="foreignCustomerType">
    <xs:sequence>
        <xs:choice>
            <xs:element name="country" maxOccurs="1" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="Sweden" />
                        <xs:enumeration value="Norway" />
                        <xs:enumeration value="Denmark" />
                        <xs:enumeration value="Finland" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element type="cre:countryWithNoSsnCheckType"
name="countryWithNoSsnCheck"
maxOccurs="1" minOccurs="0" />
        </xs:choice>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="countryWithNoSsnCheckType">
    <xs:sequence>
        <xs:element type="xs:string" name="countryCode" maxOccurs="1" minOccurs="1" />
        <xs:element type="xs:string" name="addressLine1" maxOccurs="1" minOccurs="1"
/>
        <xs:element type="xs:string" name="addressLine2" maxOccurs="1" minOccurs="1"
/>
        <xs:element type="xs:string" name="addressLine3" maxOccurs="1" minOccurs="0"
/>
        <xs:element type="xs:string" name="addressLine4" maxOccurs="1" minOccurs="0"
/>
    </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="requestType">
  <xs:sequence>
    <xs:element name="testInvoice" maxOccurs="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="true" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="makeInvoiceReservation" maxOccurs="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="1" />
          <xs:enumeration value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="forceToSend" maxOccurs="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="1" />
          <xs:enumeration value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="service" maxOccurs="1" minOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="3" />
          <xs:enumeration value="2" />
          <xs:enumeration value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="printSetup" maxOccurs="1" minOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="4" />
          <xs:enumeration value="3" />
          <xs:enumeration value="2" />
          <xs:enumeration value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="sendAs" maxOccurs="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="debtCollection" />
          <xs:enumeration value="reminder" />
          <xs:enumeration value="invoice" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="partPaymentReminderAndDebtCollectionInstallment"
maxOccurs="1" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="1" />
          <xs:enumeration value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```



```

        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:string" name="ssn" maxOccurs="1" minOccurs="1" />
    <xs:element type="xs:int" name="send_to_organization" maxOccurs="1"
minOccurs="0" />
    <xs:element type="cre:foreignCustomerType" name="foreignCustomer"
maxOccurs="1" minOccurs="0" />
    <xs:element type="cre:careOfAddressType" name="careOfAddress" maxOccurs="1"
minOccurs="0" />
    <xs:element type="xs:string" name="invoiceRef" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="invoiceOrderNo" maxOccurs="1" minOccurs="0"
/>

    <xs:element type="xs:int" name="shippingFee" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:int" name="expFee" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:int" name="invoiceDate" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:int" name="dueDate" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:float" name="overdueInterest" maxOccurs="1" minOccurs="0"
/>

    <xs:element type="xs:string" name="clientIp" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="callback" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="mobile" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="email" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="orderNo" maxOccurs="1" minOccurs="1" />
    <xs:element type="xs:string" name="ourRef" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="yourRef" maxOccurs="1" minOccurs="0" />
    <xs:element type="cre:invoiceRowsType" name="invoiceRows" />
    <xs:element type="xs:string" name="comments" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:int" name="discount" maxOccurs="1" minOccurs="0" />
    <xs:element type="xs:string" name="billingVar" maxOccurs="1" minOccurs="0" />
    <xs:element name="attachedDocument" maxOccurs="1" minOccurs="0">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1819200" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:string" name="attachedDocumentMd5" maxOccurs="1"
minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:schema>

```

3 Status Codes

| statusCode | errorCode | description |
|------------|-----------|--|
| 0 | 99 | Validate XML against XSD failed |
| 0 | 101 | ssn and orgno failed, one element should be used |
| 0 | 102 | Invalid ssn number, no match |
| 0 | 103 | Invalid organization number, no match |
| 0 | 104 | Zip code not set |
| 0 | 105 | Invalid zip |
| 0 | 106 | Phone number not set |
| 0 | 107 | Invalid phone number |
| 0 | 108 | Missing email address |
| 0 | 109 | Invalid email address |
| 0 | 110 | Missing Order Number |
| 0 | 111 | Client Ip number not set |
| 0 | 112 | Your credit is to low in this shop |
| 0 | 113 | Database error in CCM. |
| 0 | 114 | Zip and orgNo does not match |
| 0 | 115 | Zip and ssn does not match |
| 0 | 116 | Prövning på privatperson |
| 0 | 117 | Avslag på privatperson |
| 0 | 118 | Personnummer kunde inte hittas |
| 0 | 119 | ssn is not a trusted Credit customer |
| 0 | 120 | Prövning på företag |
| 0 | 121 | Avslag på företag |
| 0 | 122 | Organization number could not be found |
| 0 | 123 | Company is not a trusted Credit customer |

| statusCode | errorCode | description |
|------------|-----------|--|
| 0 | 124 | Request not received correctly |
| 0 | 125 | Key does not match |
| 0 | 10 | orderNo or OCR isn't defined |
| 0 | 40 | Quantity: To many decimals, just one is accepted! |
| 0 | 20 | Invalid ocr number |
| 0 | 21 | Invalid orderNo |
| 0 | 11 | The system is working with this invoice for the moment, please try again later |
| 0 | 26 | The invoice dueDate has past. Contact Inkassogram to redeem the invoice |
| 0 | 30 | The invoice is already redeemed |
| 0 | 31 | The invoice or some of the article rows is already redeemed |
| 0 | 13 | There's no matching article rows |
| 0 | 24 | The VAT doesn't match any rows |
| 0 | 28 | The redemption rows does not match the original invoice |
| 0 | 31 | Some of the rows is already redeemed |
| 0 | 23 | The redemption price is higher than the amount left to redeem is |
| 0 | 126 | CCM Error |
| 0 | 180 | BankID could not validate user. |
| 0 | 181 | BankID could not identify user, please try again. |
| 0 | 182 | BankID error occurred in security application, please try again, or update software. |
| 0 | 183 | BankID could not identify user, user aborted, please try again. |
| 0 | 184 | BankID could not identify user, please try again. |
| 0 | 190 | BankID could not validate user. |
| 1 | | Delivered |

4 Example Code

We have developed some code examples for 3 of the most used languages for this integration. See section 4.1, 4.2 and 4.3 below.

4.1 PHP Integration

PHP code example can be downloaded from
<https://github.com/inkassogram/api-libs/tree/master/PHP>

4.2 Java Integration

Java code example can be downloaded from
<https://github.com/inkassogram/api-libs/tree/master/Java>

4.3 .NET Integration

Java code example can be downloaded from
<https://github.com/inkassogram/api-libs/tree/master/NET>