

[NKS LTO] Partner Integration Requirements

Introduction

Native Instruments provides for all NKS partners an integration service in order to receive the customer and order data for each sale. This integration is not necessary to make sales on our side possible. It's a feature of our shop that we offer. Those messages (customer and order message) would be sent to the API endpoints the partner needs to provide. The NI team will support the partner with pleasure all the way through the integration.

General Requirements for the REST Services

The following requirements are valid for both service endpoints, the user and the order service

HTTP method	POST
Content-Type	application/json
Authentication	Basic Auth (also in form of a bearer token instead of username:password)
Protocol	HTTPS

HTTP Status Codes

All HTTP status codes apart from 200 201 and 204 will be considered as errors.

In case of an error the partner will receive an email with the error message and the payload that could not be sent. The partner has to provide this email address.

White List

The external IP/Network Addresses of our Berlin HQ and of our Data Center are [79.140.127.120/29](#), [62.214.8.48/29](#) and [85.158.3.0/26](#).

Dev and Prod Environments

It would be great if besides the live API points the partner provides also test API points.

Response Message

There are no specific requests apart from using code 200 in case of success

User Information REST Service

JSON Request Message

```
{
  "source" : "Native Instruments Webshop",
  "customerDetails" : {
    "sdfsUserId" : "3605000",
    "business" : false,
    "firstName" : "German",
    "lastName" : "Customer1",
    "emailAddress" : "german_customer1@s.ni.de",
    "phoneNumber" : "030 3480324803",
    "language" : "en_US",
    "vatId" : null,
    "category" : "DIRECT SALES"
  },
  "addresses" : {
    "address" : [ {
      "location" : {
        "firstName" : "German",
        "lastName" : "Customer1",
        "companyName" : null,
        "address1" : "Deutsche Str. 1",
        "address2" : null,
        "address3" : null,
        "city" : "Berlin",
        "state" : null,
        "zipCode" : "10244",
        "county" : null,
        "province" : null,
        "country" : "DE",
        "addressee" : null,
        "addressType" : "BILL_TO"
      }
    }, {
      "location" : {
        "firstName" : "German",
        "lastName" : "Customer1",
        "companyName" : null,
```

```

        "address1" : "Deutsche Str. 1",
        "address2" : null,
        "address3" : null,
        "city" : "Berlin",
        "state" : null,
        "zipCode" : "10244",
        "county" : null,
        "province" : null,
        "country" : "DE",
        "addressee" : null,
        "addressType" : "SHIP_TO"
    }
} ]
}
}

```

Corresponding XML Schema

```

<?xml version= '1.0' encoding= 'UTF-8' ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns="http://xmlns.native-instruments.de/SOA/NKS/Partner/Custom
er/V1"

targetNamespace="http://xmlns.native-instruments.de/SOA/NKS/Partner/Custom
er/V1"

    elementFormDefault="qualified">
    <xsd:element name="customer" type="customerType"/>
    <xsd:complexType name="customerType">
    <xsd:sequence>
    <xsd:element name="source" type="xsd:string"/>
    <xsd:element name="customerDetails">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="sdfsUserId" type="xsd:string"/>
    <xsd:element name="business" type="xsd:boolean"/>
    <xsd:element name="firstName" type="xsd:string"/>
    <xsd:element name="lastName" type="xsd:string"/>
    <xsd:element name="emailAddress" type="xsd:string"/>
    <xsd:element name="phoneNumber" type="xsd:string"/>
    <xsd:element name="language" type="xsd:string"/>
    
```

```

        <xsd:element name="vatId" type="xsd:string" nillable="true"/>
        <xsd:element name="category" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="addresses">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="address" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="location">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:element name="firstName" type="xsd:string"/>
                                    <xsd:element name="lastName" type="xsd:string"/>
                                    <xsd:element name="companyName"
type="xsd:string"/>
                                    <xsd:element name="address1" type="xsd:string"/>
                                    <xsd:element name="address2" type="xsd:string"/>
                                    <xsd:element name="address3" type="xsd:string"/>
                                    <xsd:element name="city" type="xsd:string"/>
                                    <xsd:element name="state" type="xsd:string"/>
                                    <xsd:element name="zipCode" type="xsd:string"/>
                                    <xsd:element name="county" type="xsd:string"/>
                                    <xsd:element name="province" type="xsd:string"/>
                                    <xsd:element name="country" type="xsd:string"/>
                                    <xsd:element name="addressee" type="xsd:string"/>
                                    <xsd:element name="addressType"
type="xsd:string"/>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>

```

```
</xsd:complexType>
</xsd:schema>
```

Order Information REST Service

JSON Request Message

2 different products with quantity 1

```
{
  "customer" : {
    "email" :
"soa-dev-errors+german_customer1@native-instruments.de",
    "firstName" : "German",
    "lastName" : "Customer1",
    "language" : "en"
  },
  "order" : {
    "orderID" : "12591040",
    "customerReference" : "GermanCustDLNoErrors_1594062693374",
    "item" : [ {
      "skuNumber" : "27224",
      "description" : "Spitfire Audio - Hauschka Composer Toolkit NKS
Store",
      "serialNumbers" : [ {
        "serialNumber" : "W01K2-35272-8UE25-437B9-D57K6",
        "serialReference" : "W0120356362968"
      } ]
    }, {
      "skuNumber" : "27227",
      "description" : "Spitfire Audio - Olafur Arnalds Chamber Evolutions
NKS Store",
      "serialNumbers" : [ {
        "serialNumber" : "P78CB-7998F-C444E-4ED49-3U3D7",
        "serialReference" : "P7820356362975"
      } ]
    } ]
  }
}
```

JSON Request Message

2 different products with quantity 2

```
{
  "customer" : {
    "email" :
"soa-dev-errors+german_customer1snide@native-instruments.de",
    "firstName" : "German",
    "lastName" : "Customer1",
    "language" : "en"
  },
  "order" : {
    "orderID" : "12591045",
    "customerReference" : "GermanCustDLNoErrors_1594114598842",
    "item" : [ {
      "skuNumber" : "27224",
      "description" : "Spitfire Audio - Hauschka Composer Toolkit NKS
Store",
      "serialNumbers" : [ {
        "serialNumber" : "W01F2-E434E-823E4-4KD7A-3D954",
        "serialReference" : "W0120356363002"
      }, {
        "serialNumber" : "W01D2-44UEK-CEC9A-44A5B-6C67B",
        "serialReference" : "W0120356363019"
      } ]
    }, {
      "skuNumber" : "27227",
      "description" : "Spitfire Audio - Olafur Arnalds Chamber Evolutions
NKS Store",
      "serialNumbers" : [ {
        "serialNumber" : "P78BF-CAF9K-B5EEU-4U7F8-7FKU2",
        "serialReference" : "P7820356363026"
      }, {
        "serialNumber" : "P78B3-A49AA-DUE63-4958B-825U2",
        "serialReference" : "P7820356363033"
      } ]
    } ]
  } ]
}
```

```
}
```

Corresponding XML Schema

```
<?xml version= '1.0' encoding= 'UTF-8' ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns="http://xmlns.native-instruments.de/SOA/NKS/Partner/Registration/V1"

targetNamespace="http://xmlns.native-instruments.de/SOA/NKS/Partner/Regist
ration/V1"

    elementFormDefault="qualified">
    <xsd:element name="downloadOrderNotification"
type="downloadOrderNotificationType"/>
    <xsd:complexType name="downloadOrderNotificationType">
    <xsd:sequence>
    <xsd:element name="customer">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="email" type="xsd:string"/>
    <xsd:element name="firstName" type="xsd:string"/>
    <xsd:element name="lastName" type="xsd:string"/>
    <xsd:element name="language" type="xsd:string"/>
    </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
    <xsd:element name="order">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="orderId" type="xsd:string"/>
    <xsd:element name="customerReference" type="xsd:string"/>
    <xsd:element name="item" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="skuNumber" type="xsd:string"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="serialNumbers" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:sequence>
```

```

        <xsd:element name="serialNumber"
type="xsd:string"/>
        <xsd:element name="serialReference"
type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

SDBS Bundle (Real Bundle) vs Virtual Bundle

There are 2 possible Bundle setups:

SDBS Bundle (Real Bundle) - the serial registration JSON would contain only one SKU for the main item with only one Serial Number that is valid for all the bundle components (bundle components not in the JSON)

Virtual Bundle - the serial registration JSON would contain only the bundle components with their corresponding serial numbers

Item mapping

The SKUs in the order message sent can be ours or the SKUs of the partner, If the partner wants us to use theirs, we need the partner's SKUs to match them to ours.