

# **WinOrder-Dokumentation:**

## **WinOrder EShop-Spezifikation**

Version 1.8.14  
Stand: 10.04.2024

## Inhalt

Änderungen .....	3
Einleitung .....	5
Übertragungsmöglichkeiten .....	5
Ablage der Bestelldatei .....	6
Verarbeitung der Bestelldatei .....	6
Fehlerbehandlung .....	6
Format der Bestelldatei .....	6
XML-Formatbeschreibung .....	10
JSON Format .....	15
Zuordnung des Kunden .....	17
Zuordnung der Artikel .....	17
Synchronisation von Artikeln .....	18
EShop Einstellungen vornehmen .....	19
Testen des EShops .....	19
Export der Artikeldaten .....	20
E-Mail-Schnittstelle .....	22
Webservice Schnittstelle (REST-API) .....	23
Webservice Schnittstelle (SOAP) .....	27
WSDL Definition der Schnittstelle .....	29
Echtzeitübermittlung mit Pusher-Channels: .....	31
Filialweiterleitung / CallCenter Funktion (WinOrder 8.0 Enterprise) .....	32
Eingebauter Webserver (WinOrder 8.0) .....	32
Beispielimplementierung in PHP .....	34

## Änderungen

11.09.2007	<Comment> durch <DescriptionOfWay> ersetzt. Die ist die Wegbeschreibung des Kunden.
26.09.2007	<Article> Einfügen als neuer unbekannter Artikel oder als Kommentar.
15.08.2008	ArticleNo als alphanumerischer Wert CURRENCY Typ klargestellt <OrderID> für Bestellungen eingeführt
19.12.2008	Webservice Schnittstelle eingeführt
21.09.2009	<OrderID> ist alphanumerisch
06.01.2011	<Partition> korrigiert
18.01.2010	Schreibfehler korrigiert
07.06.2011	<ServerData> und <DiscountValue> korrigiert
05.12.2013	Aktualisierung für WinOrder 4.0
26.03.2014	XML-Datumswerte klargestellt
10.06.2014	Artikelsynchronisierung erweitert
21.07.2015	Datum-Uhrzeitformat klargestellt
18.10.2016	Zahlartgebühr <PaymentFee> eingeführt Trinkgeld <Tip> eingeführt
06.12.2016	<EShopName> eingeführt für Zuordnung zu WinOrder Online-Shop Einstellung <AcceptedByEmployeeNo> eingeführt für die Zuordnung des Mitarbeiters der Bestellaufnahme <PayPalEmail> eingeführt
27.01.2017	<TransactionID> für Bezahlvorgänge eingeführt
02.06.2017	<NoMarketing> eingeführt
12.10.2017	<Total> Summe des Online-Shop eingeführt <MinOrderValue>, <MinQuantitySurcharge> Mindestbellwert / Mindermengenaufschlag hinzugefügt

05.12.2017	Erweiterungen des Tracking-Status Es werden Unix-Timestamps in Datum/Zeit Felder akzeptiert
07.12.2017	
15.10.2018	REST-API dokumentiert
18.06.2019	Aktualisierung für WinOrder 6.0 Tracking-Status erweitert um Status „10“=„storniert“
22.10.2019	
05.02.2020	<DiscountName> eingeführt Für Abholer- und Tischbestellungen erweitert
16.06.2020	
17.11.2020	<ContactlessDelivery> eingeführt Pusher Echtzeitübertragung dokumentiert
10.03.2021	
13.04.2021	Dine-In Funktion (Tischbestellungen) eingeführt <PromisedETA> Durch Online-Shop/App zugesagte Lieferzeit
18.10.2021	
25.03.2022	Aktualisierung für WinOrder 7.0 PHP-Server Beispiel auf Github
22.07.2022	
08.03.2023	Aktualisierung für WinOrder 8.0 GEO-Position für Kunden (Latitude/ Longitude) dokumentiert
10.05.2023	
10.04.2024	Aktualisierung für WinOrder 9.0

## Einleitung

WinOrder besitzt eine Schnittstelle zur Übernahme von Internet-Bestellungen. Die Bestellungen werden durch den Webshop generiert, per E-Mail, HTTP/JSON, REST-API oder SOAP auf den lokalen Rechner übertragen und durch WinOrder eingelesen. Die Datei mit den Bestelldaten wird in einem Warteschlange-Ordner abgelegt und durch den integrierten Shop-Moderator abgearbeitet. Der Shop-Moderator erlaubt die Sichtung der Bestelldaten und das Auflösen von Konflikten (Adress- und Artikeldaten) durch den WinOrder Benutzer.

## Übertragungsmöglichkeiten

WinOrder unterstützt mehrere Verfahren, die Bestellung auf den lokalen Rechner zu übertragen:

- Das Empfangen aus einem E-Mail-Postfach: WinOrder überprüft in einem bestimmten Intervall das Postfach auf neue Nachrichten und ruft die Bestellungen automatisch ab.
- Der Shop-Betreiber übernimmt selbst die Übertragung und legt die Bestelldaten in dem von WinOrder überwachten Verzeichnis „%PROGRAMDATA%\PixelPlanet\WinOrder6\EShop\Incoming“ ab.
- Übertragung mittels REST-API /JSON Schnittstelle Bei dieser Übertragungsart kann WinOrder eine Rückmeldung über den aktuellen Verarbeitungsstatus sowie aktuelle geschätzte Lieferzeit an den Online-Shop zurücksenden (Tracking-Status)
- Übertragung mittels Webservice/SOAP Schnittstelle. Bei dieser Übertragungsart kann WinOrder eine Rückmeldung über den aktuellen Verarbeitungsstatus an den Online-Shop zurücksenden (Tracking-Status)
- POST an den in WinOrder eingebauten Webserver.
- **Veraltet:** Die direkte Übertragung per ISDN: Der Bestellservers wählt sich auf den Rechner per ISDN ein und überspielt die Bestellung direkt in den Warteschlangenordner. Dieses Verfahren hat den Vorteil, dass die Bestellung sofort überspielt werden kann und der Kunde eine definierte Rückmeldung über den erfolgreichen Versand bekommt.

## **Ablage der Bestelldatei**

WinOrder erwartet neue EShop-Bestellungen im überwachten Verzeichnis (Windows XP)  
*C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\PixelPlanet\WinOrder6\EShop*

bzw. für Windows Vista, 7 und 8 im Verzeichnis  
*C:\ProgramData\PixelPlanet\WinOrder6\EShop*

Das Basisverzeichnis ist der Windows-App-Settings Ordner für alle Benutzer  
*CSIDL\_COMMON\_APPDATA + ,WinOrder\EShop'*

Auf diesem Verzeichnis wacht ein Datei-Monitor, der das Programm über neu eingetroffene Bestellungen informiert und den Shop-Moderator startet. Damit ist es möglich benutzerdefinierte Übertragungsarten zu implementieren. Der Dateiname muss eindeutig gewählt werden, damit nicht schon vorhandene Bestellungen überschrieben werden. Als Dateiendungen werden „.XML“ für unkomprimierte und „.ZLIB“ für komprimierte Daten (.Z / GZip Dateien) erwartet. Für JSON codierte Dateien kann die Dateiendung „.json“ verwendet werden.

Andere Dateitypen werden nicht verarbeitet und verbleiben unverändert im Verzeichnis.

## **Verarbeitung der Bestelldatei**

Wird eine Bestelldatei mit der Dateiendung „.XML“ „.JSON“ oder „.ZLIB“ in den Unterordner „\EShop\Incoming“ kopiert, wird automatisch der Shop-Moderator gestartet. Nach Übernahme in die Software wird die Datei in den Unterordner „\EShop\Processed“ verschoben. Damit ist es möglich bereits abgearbeitete Bestellungen weiter zu verarbeiten bzw. zu archivieren.

## **Fehlerbehandlung**

Bei fehlerhafter Verarbeitung wird eine „stille Exception“ ausgelöst, d.h. der Benutzer bekommt keine direkte Fehlermeldung als Dialog. Stattdessen erscheint in der unteren rechten Ecke ein Ausrufezeichen. Bei Klick auf das Icon wird die Fehlermeldung angezeigt. Des Weiteren lassen sich Fehler in der Diagnose anzeigen (Hilfe/WinOrder-Support/Diagnose).

## **Format der Bestelldatei**

Die Bestelldatei kann komprimiert oder unkomprimiert übertragen werden. Derzeit wird die Kompressionsart ZLIB unterstützt. Durch die Dateiendungen erkennt WinOrder, ob es sich um eine komprimierte Datei handelt und entpackt die Daten vor der Verarbeitung. Durch die Kompression kann die Übertragungszeit reduziert werden. Nach der Dekompression liegt die Bestellung als XML vor:

### XML-Bespieldatei:

```
<WinOrder>
  <OrderList>
    <CreateDateTime>12.10.2013 13:22:04</CreateDateTime>
    <Order>
      <OrderID>21</OrderID>
      <Customer>
        <CustomerNo>25</CustomerNo>
        <DeliveryAddress>
          <Title>Herr</Title>
          <FirstName>Erika</FirstName>
          <LastName>Mustermann</LastName>
          <Company>PixelPlanet GmbH</Company>
          <Street>Hoyaer Str</Street>
          <HouseNo>13</HouseNo>
          <AddAddress>Am Brommyplatz</AddAddress>
          <DescriptionOfWay>Optionale Wegbeschreibung</DescriptionOfWay>
          <Zip>28205</Zip>
          <City>Bremen</City>
          <State>Bremen</State>
          <Country>Deutschland</Country>
          <Email>info@pixelplanet.de</Email>
          <PayPalEmail >info@pixelplanet.de</PayPalEmail >
          <PhoneNo>0421/247780</PhoneNo>
          <Fax>0421/2477824</Fax>
        </DeliveryAddress>
        <BillAddress>
          <Title></Title>
          <FirstName></FirstName>
          <LastName></LastName>
          <Company></Company>
          <AddAddress></AddAddress>
          <Street></Street>
          <HouseNo></HouseNo>
          <Zip></Zip>
          <City></City>
          <State></State>
          <Country></Country>
          <Email></Email>
          <PhoneNo></PhoneNo>
          <Mobile></Mobile>
          <Fax></Fax>
          <Comment></Comment>
        </BillAddress>
      </Customer>
      <ServerData>
        <IpAddress>127.0.0.1</IpAddress>
```

```
<Agent>Mozilla Firefox</Agent>
<Referer>http://www.winorder.de</Referer>
<CreateDateTime>12.10.2013 13:22:00</CreateDateTime>
</ServerData>
<StoreData>
  <StoreId>1</StoreId>
  <StoreName>PixelPlanet Testshop</StoreName>
</StoreData>
<AddInfo>
  <DateTimeOrder>15:00:00</DateTimeOrder>
  <DiscountValue>2</DiscountValue>
  <CurrencyStr>€</CurrencyStr>
  <DeliverLumpSum>5.00 €</DeliverLumpSum>
  <DeliverType>Lieferung</DeliverType>
  <AcceptedByEmployeeNo>-1</ AcceptedByEmployeeNo >
  <DeliveredByEmployeeNo>-1</DeliveredByEmployeeNo>
  <Comment>Fahrer soll hupen</Comment>
  <PaymentType>Online bezahlt</PaymentType >
  <PaymentFee>1.20 €</PaymentFee>
  < MinOrderValue>20.00 €</ MinOrderValue >
  < MinQuantitySurcharge>1.00 €</ MinQuantitySurcharge >
  < TransactionID >ID0815</ TransactionID >
  <Tip>1.00 €</Tip>
  <Total>11.00 €</Total>
</AddInfo>
<ArticleList>
  <Article>
    <ArticleNo>23A</ArticleNo>
    <ArticleName>Grundpizza</ArticleName>
    <ArticleSize>klein (26 cm)</ArticleSize>
    <Count>1</Count>
    <Price>5.9</Price>
    <Tax></Tax>
    <Deposit></Deposit>
    <Comment></Comment>
    <SubArticleList>
      <SubArticle>
        <ArticleNo></ArticleNo>
        <ArticleName></ArticleName>
        <Count></Count>
        <Price></Price>
        <Partition></Partition>
        <Tax></Tax>
        <Deposit></Deposit>
        <Comment></Comment>
      </SubArticle>
      ...
    </SubArticleList>
```



```
</Article>
...
</ArticleList>
</Order>
...
</OrderList>
</WinOrder>
```

## XML-Formatbeschreibung

### Header:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Enthält XML-Versionsnummer und verwendete Enkodierung. Bevorzugt wird UTF-8 (Unicode), möglich sind aber auch andere Kodierungen wie z.B. ISO-8859 (westlicher Zeichensatz). Datum/Uhrzeitwerte können mit aktuellem Länderschema, im US-Format oder am besten als gültige XML Datum/Uhrzeit angegeben werden.

Beachten Sie die richtige Kodierung vor allem von Sonderzeichen. Zur Fehlersuche können Sie die Datei mit dem Internet-Explorer öffnen. Hier muss sie richtig angezeigt werden

### Datenteil:

```
<WinOrder>...</WinOrder>
```

Dieser Bereich umfasst alle relevanten Teile der Bestellung. XML ist **Case-Sensitive**, beachten Sie also die genaue Schreibweise (Groß- Kleinschreibung) der Tags/Entitäten. Beachten Sie auch die Formatierung von Gleitkommazahlen / Währungswerte im US-Format (7.2 anstelle von 7,4).

```
<OrderList>...</OrderList>
```

Dieser Bereich enthält die gesamte Bestellliste. Eine XML-Datei kann auch mehrere Bestellungen enthalten. Damit ist es möglich mehrere Bestellungen auf dem Server zu sammeln und als ein Paket zu versenden.

```
<Order>...</Order>
```

Dieser Bereich enthält eine Bestellung.

Name:	Format:	Beschreibung:
OrderID	STRING	Eindeutige ID für die Bestellung. Kann zur Status-Rückmeldung an den Server verwendet werden. Alphanumerische Werte sind zugelassen.

```
<Customer>...</Customer>
```

Dieser Bereich enthält alle Kundeninformationen. Aufgeteilt in Rechnungs- und Lieferadresse wird derzeit nur die Lieferadresse verarbeitet und die Rechnungsadresse als Kommentar in die Bestellung eingefügt. Derzeit wird nur die Lieferadresse berücksichtigt.

Name:	Format:	Beschreibung:
CustomerNo	INTEGER	Kundennummer, Tischnummer bei Dine-In
Title	ENUM	Anrede („Herr“, „Frau“, „Firma“, „Familie“)
FirstName	STRING	Vorname
LastName	STRING	Nachname
Company	STRING	Firma
Street	STRING	Straße
HouseNo	STRING	Hausnummer
AddAddress	STRING	Adresszusatz (Stockwerk etc.)
Zip	STRING	Postleitzahl
DescriptionOfWay	STRING	Optionale Wegbeschreibung
City	STRING	Ort
State	STRING	Bundesland
Country	STRING	Land
E-Mail	STRING	E-Mail-Adresse
PayPalEmail	STRING	PayPal E-Mail-Adresse (z.B. für Zahlungsanforderungen)
PhoneNo	STRING	Telefonnummer
Mobile	STRING	Handynummer
Fax	STRING	Faxnummer
Latitude	REAL	GEO-Position des Kunden (Breitengrad)
Longitude	REAL	GEO-Position des Kunden (Längengrad)
NoMarketing	STRING	„1“ oder „true“ wenn Kunde keine Werbung wünscht (z.B. Newsletter/Serienbrief)

<ServerData>...</ServerData>

Dieser Bereich enthält die Serverinformationen wie IP-Adresse und Browser Kennung des Bestellers:

Name:	Format:	Beschreibung:
IpAddress	STRING	IP-Adresse des Bestellers. Kann als Kommentar in die Bestellung eingefügt werden.
Agent	STRING	Browserkennung, z.B. „Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)“
CreateDateTime	TIMESTAMP	Datum und Uhrzeit des Bestelleingangs auf dem Server. Kann als Kommentar in die Bestellung eingefügt werden. Die Zeit sollte als gültige XML-Zeit mit Zeitzone angegeben werden, z.B. 2014-02-25T12:00:00.000+01:00 für den 25.2.2014 zu 12:00 lokaler Zeit

--	--	--

<StoreData>...</StoreData>

Dieser Bereich enthält die Informationen zur gewählten Filiale:

Name:	Format:	Beschreibung:
StoreID	INTEGER	Eindeutige Nummer der Filiale. Optional
StoreName	STRING	Name der Filiale. Optional

<AddInfo>...</AddInfo>

Dieser Bereich enthält zusätzliche Informationen zur Bestellung:

Name:	Format:	Beschreibung:
DateTimeOrder	DATETIME	Datum und Uhrzeit bei Zeit-/Vorbereitung, sonst nicht angegeben. Darf nicht in der Vergangenheit oder > 2 Monate in Zukunft liegen.
DiscountPercent	REAL	Rabatt in Prozent. Im Normalfall nicht angegeben oder 0. Wird ein Rabatt in Prozent angegeben muss ebenfalls der Rabattwert definiert sein. Bereich 0..100%
DiscountValue	CURRENCY	Absoluter Rabattwert.
DiscountName	STRING	Name des Rabatts, z.B. „Rabatt“, „Couponeinlösung“
ContactlessDelivery	BOOL	Kommentar für kontaktlose Lieferung. Fahrer soll Ware vor der Tür abstellen. Wird als erster Kommentar eingefügt und fett dargestellt.
CurrencyStr	STRING	Optionales Währungssymbol
DeliverLumpSum	CURRENCY	Lieferpauschale
DeliverType oder DeliveryType	ENUM	Lieferart („Lieferung“, „Abholer“/„Abholung“, „pickup“, „take-away“, „dine-in“)
AcceptedByEmployeeNo	Integer	Durch diesen Mitarbeiter wurde die Bestellung aufgenommen/vermittelt.
DeliveredByEmployeeNo	Integer	Bestellung wird automatisch auf Mitarbeiter gebucht. Hier muss die Mitarbeiter-Nummer angegeben werden.
Comment	STRING	Kommentar des Kunden zur Bestellung (z.B.: „Fahrer soll bei Ankunft hupen.“)
PaymentType	STRING	Zahlungsart, z.B. „Online bezahlt“. Der Text muss mit einem Wert der Stammdaten / Zahlungsarten übereinstimmen
PaymentFee	CURRENCY	Zahlartgebühr
MinQuantitySurcharge	CURRENCY	Mindermengenaufschlag (Unterschreitung des Mindestbestellwertes)

MinOrderValue	CURRENCY	Mindestbestellwert
Tip	CURRENCY	Trinkgeld
TransactionID	STRING	Transaktions-ID für Bezahlvorgänge, wird auf der Bestellung ausgegeben
PromisedETA	DATETIME	Falls der Online-Shop eine zugesagte Lieferzeit für den Kunden setzt kann der Lieferzeitpunkt hier übertragen werden
Total	CURRENCY	Die zu zahlende Gesamtsumme (berechnet durch den Shop).

`<ArticleList>...</ArticleList>`

Dieser Bereich enthält die bestellten Artikel.

`<Article>...</Article>`

Ein bestellter Artikel:

Name:	Format:	Beschreibung:
ArticleNo	INTEGER / STRING	Eindeutige Artikelnummer
ArticleName	STRING	Name des Artikels.
ArticleSize	STRING	Größenangabe zum Artikel
Count	INTEGER	Anzahl
Price	CURRENCY	Preis des Artikels
Tax	REAL	Mehrwertsteuer in Prozent
Deposit	CURRENCY	Optionaler Pfandanteil
Comment	STRING	Optionaler Kommentar

Ein bestellter Artikel kann zusätzlich eine Liste von Zusätzen enthalten. Diese Zusatzartikel/Toppings können nur dann bestellt werden, wenn zuvor ein Hauptartikel ausgewählt wurde (z.B. Dressing zum Salat, Salami auf Pizza):

`<SubArticleList>...</SubArticleList>`

`<SubArticle>...</SubArticle>`

Bestellter Zusatzartikel, der abhängig vom bestellten Hauptartikel ist:

Name:	Format:	Beschreibung:
ArticleNo	INTEGER / STRING	Eindeutige Artikelnummer
ArticleName	STRING	Name des Artikels.
Count	INTEGER	Anzahl
Partition	ENUM	Optionale Aufteilung nach Hälften („<“, „>“, „+“, „-“, „--“)
Price	CURRENCY	Preis des Artikels

Tax	REAL	Mehrwertsteuer in Prozent
Deposit	CURRENCY	Optionaler Pfandanteil
Comment	STRING	Optionaler Kommentar

Ist die Artikelnummer und der Artikelname nicht angegeben wird der Zusatzartikel als Kommentar (Unbekannter Artikel) eingefügt.

Falls Artikel-Nr. und Name nicht- der Kommentar jedoch angegeben ist, wird der Artikel als Kommentar eingefügt, nicht jedoch in die Liste unbekannter Artikel aufgenommen. Damit kann verhindert werden, dass Kommentare als neue EShop Artikel angelegt werden.

Feldtypen:

Feldtyp:	Format:
INTEGER	Ganzzahl
STRING	Text
REAL	Gleitkommazahl mit Punkt als Dezimaltrennzeichen (US-Format)
CURRENCY	Währungsfeld in der Form „1,25 €“, „1,25“ oder „1.25“
DATETIME	Datum-/Zeitstempel in der im Betriebssystem eingestellten Form „hh:mm:ss dd.mm.yy“ oder US-Datumsformat
ENUM	Aufzählungstyp

## JSON Format

Die Beispiel XML-Bestellung im JSON-Format:

```
{
  "OrderList": {
    "Order": {
      "AddInfo": {
        "DiscountValue": "2",
        "Comment": "OnlineShop-Bestellung vom 12.10.2013 13:22:00 (IP-
Adresse: 127.0.0.1)",
        "DeliverLumpSum": "5",
        "CurrencyStr": "\u00e2\u201a\u00ac",
        "PaymentType": "Barzahlung"
      },
      "ArticleList": {
        "Article": [
          {
            "Comment": "Fahrer soll hupen",
            "Price": "0",
            "Count": "1"
          },{
            "Price": "5.9",
            "ArticleSize": "klein (26 cm)",
            "ArticleName": "Grundpizza",
            "ArticleNo": "1",
            "Count": "1"
          },{
            "Comment": "OnlineShop-Bestellung vom 12.10.2013 13:22:00 (IP-
Adresse: 127.0.0.1)",
            "Price": "0",
            "Count": "1"
          },{
            "Comment": "Lieferung",
            "Price": "0",
            "Count": "1"
          }
        ]
      },
      "StoreData": {
        "StoreId": "1",
        "StoreName": "WinOrder Demofiliale"
      },
      "ServerData": {
        "Agent": "WinOrder V6.0.0.0",
        "CreateDateTime": "2014-01-29T17:04:37.470+01:00",
        "Referer": "WinOrder",
        "IpAddress": "192.168.1.112"
      },
      "Customer": {
        "DeliveryAddress": {
          "PhoneNo": "--"
        },
        "CustomerNo": "-1"
      }
    }
  },
}
```

```
"CreateDateTime": "2014-01-29T17:04:37.469+01:00"  
}  
}
```



## Zuordnung des Kunden

WinOrder versucht den angegebenen Kunden in der Datenbank zu finden und zuzuordnen. Dabei wird folgende Suchreihenfolge verwendet:

1. Kundennummer. Wird ein Lieferkunde (oder Tisch, s. Punkt 6) mit der angegebenen Kundennummer gefunden, wird die Bestellung diesem Kunden zugeordnet.
2. Telefonnummer, Nachname, Straße, Hausnummer, Ort. Wird ein Kunde mit der angegebenen Telefonnummer gefunden vergleicht WinOrder die restlichen Daten (ohne Berücksichtigung der Groß- / Kleinschreibung). Werden mehrere Telefonnummern angegeben (Semikolon getrennt), wird nur die erste Telefonnummer zur Suche verwendet.  
Wird ein anderer Kunde mit gleicher Telefonnummer gefunden, erscheint im EShop vor dem Kunden ein gelbes Ausrufezeichen. Bei der Bestellaufnahme wird der Anwender zum Vergleich der Kundendaten aufgefordert.
3. Nachname, Straße, Hausnummer, Ort. Wird weder Kundennummer noch Telefonnr. angegeben werden nur die verbleibenden Werte zur Suche herangezogen.
4. <DeliverType> = Abholung, Abholer, pickup oder take-away: Es wird der Abholerkunde verwendet.
5. <LastName> = Tischname ohne weitere Kundendaten: Es wird der Tischkunde verwendet.
6. DeliveryType= „dine-in“: Eine Tischnummer wird über CustomerNo gesucht.

Wird der Kunde gefunden, erscheint im Shop-Moderator die Kundennummer und ein grünes Häkchen vor der Bestellung. Andernfalls wird eine in WinOrder zu dem Shop vorgenommene Einstellung ausgewertet, ob der unbekannte Kunde automatisch als neuer Kunde angelegt werden soll. Anderenfalls fragt das Programm mit Übernehmen den Anwender nach Anlage eines neuen Kunden. Wird ein neuer Kunde gespeichert, kehrt das Programm in den Shop-Moderator zurück. Diesmal erscheint vor der Bestellung das grüne Häkchen. Bei einer Übereinstimmung der Telefonnummer aber Abweichung des Namens bzw. der Adresse erscheint das gelbe Ausrufezeichen.

## Zuordnung der Artikel

WinOrder versucht die bestellten Artikel in der Datenbank zu finden und innerhalb der Bestellannahme zuzuordnen:

1. Bei Angabe einer Artikelnummer wird diese verwendet.
2. Wurde keine Artikelnummer angegeben oder nicht in der Datenbank gefunden wird der Artikelname zur Suche verwendet. Dabei erfolgt keine Berücksichtigung der Groß- / Kleinschreibung.

Falls kein entsprechender Artikel gefunden wurde verwendet WinOrder die Kommentar Funktion zum Hinzufügen. Vor dem Kommentar wird anstelle des Artikelsymbols ein Ausrufezeichen verwendet.

Es wird empfohlen die Artikeldaten des Shops mit den Artikeldaten von WinOrder zu synchronisieren, da ansonsten die Bestellung nur aus Kommentaren besteht. Bei Kommentaren können keine Artikelumsätze gebildet werden und eine Änderung von Bestelldaten ist nicht ohne weiteres möglich.

## Synchronisation von Artikeln

Erstellen Sie im Shop eine Bestellung mit allen unbekannten Artikeln und schicken diese ab. Beim Import über den EShop-Moderator fragt Sie WinOrder ob die unbekannten Artikel in den Artikelstamm übernommen werden sollen. Um das automatische Hinzufügen von unbekannten Artikeln zu aktivieren, stellen Sie unter Optionen/Einstellungen/EShop „Bei unbekanntem Artikel warnen“ auf „Ja“:

The screenshot shows the WinOrder 3.0 Professional-Version EShop Moderator interface. A confirmation dialog titled "Bestätigung" is displayed over a table of order items. The dialog asks: "1 Artikel wurden nicht im Artikelstamm gefunden. Möchten Sie die fehlenden Artikel jetzt dem Artikelstamm hinzufügen?" with "Ja" and "Nein" buttons.

**Verfügbare EShop-Bestellungen:**

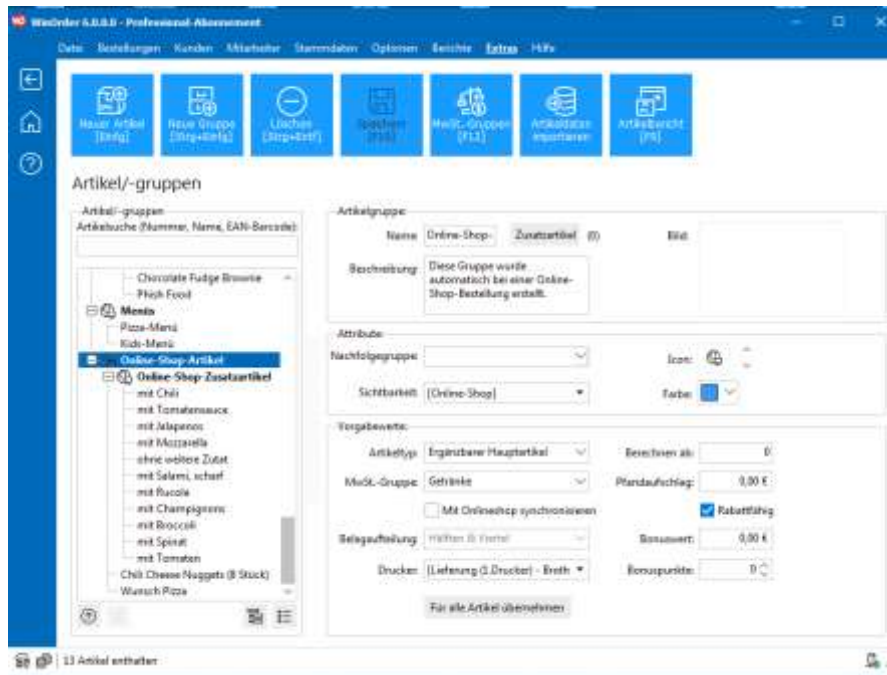
K.Nr.	Bestellzeit	Besteller	Adresse	Ort	Telefon
??	16.05.2006, 17...	Krankenhaus1	Krankenhausweg 4-19	Musterstadt	0421

1 Bestellung

**Bestelldetails:**

Art.Nr.	Bezeichnung	Preis	Preis gesamt
1	Internet-Grundpizz	3,40 €	3,40 €
62	Schink	0,50 €	0,50 €
63	Puten	0,40 €	0,40 €
66	Mais	0,60 €	1,20 €
331	Coca Cola 0,5l (1,10)	1,00 €	1,00 €
	Pfand	0,15 €	0,15 €
101	Hor muk gai	11,70 €	11,70 €
	Extra Schale mit Gurken	2,50 €	2,50 €
Endsumme			20,85 €
1 Artikel nicht im Artikelstamm gefunden.			20,85 €

Klicken Sie auf „Ja“ um die unbekanntem Artikel in die Stammdaten zu übernehmen. Hinweis: Nicht alle Eigenschaften können aus dem EShop übernommen werden, eine manuelle Nachbearbeitung ist notwendig. Unter Stammdaten/Artikel finden Sie eine neue Artikelgruppe „EShop Artikel“ mit den unbekanntem Artikeln:



Ergänzen Sie nun die fehlenden Angaben wie z.B. Artikelnummer oder MwSt.-Gruppe und speichern die Änderungen.

Bei der nächsten EShop-Bestellung werden die Artikel dann korrekt erkannt.

## EShop Einstellungen vornehmen

Informationen zur Einrichtung finden Sie in der Online-Hilfe bzw. im Handbuch. Schauen Sie auch unter „Optionen“ / „Einstellungen“ / „EShops“

## Testen des EShops

Basisverzeichnis für das überwachte EShop-Verzeichnis in Windows XP ist

*C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\PixelPlanet\WinOrder6\EShop*

bzw. ab Windows Vista

*C:\ProgramData\PixelPlanet\WinOrder\EShop*

Im Unterordner „EShop\Processed“ finden Sie eine Beispiel-Bestellung. Kopieren Sie diese Datei in den Eingangsordner „EShop\Incoming“, der Shop-Moderator wird sofort gestartet.

Sie können auch eine Bestellung als EShop-Bestellung speichern. Wählen Sie dazu im Hauptmenu der Bestellaufnahme „Als EShop-Bestellung speichern“. Die generierte XML/JSON Datei kann dann in den Ordner „Incoming“ verschoben werden.

## Export der Artikeldaten

WinOrder bietet verschiedene Möglichkeiten die Artikeldaten zu exportieren: natives Format, CSV, XML und Pizza.de/Bringdienst XML und JSON.

Das native Format dient zum Austausch von Artikeldaten zwischen zwei WinOrder Programmen, z.B. um Artikel in eine neue Filiale zu importieren. Es ist nicht weiter dokumentiert.

Das CSV-Format dient zum einfachen Export als Textdatei und kann z.B. mit MS-Excel angezeigt werden. Folgende Felder stehen zur Verfügung:

Name:	Beschreibung:
CAPTION	Name des Artikels.
DESCRIPTION	Beschreibung des Artikels.
ARTICLENO	Eindeutige Artikelnummer
ARTICLEGROUPREF	Referenz auf die Artikelgruppe
DEFAULTPRICE	Standard-Preis des Artikels
GROUPNAME	Name der Artikelgruppe
PARENTGROUPNAME	Name der übergeordneten Artikelgruppe
WAREKINDGROUP	Referenz auf Warengruppe

Das XML-Format bietet eine hierarchische Darstellung der Artikeldaten.

Es gibt die Entitäten <TGroupData> und <TArticleData>, die jeweils Artikel und Gruppen repräsentieren.

Beschreibung der Daten für <TGroupData>:

Name:	Beschreibung:
<Caption>	Name der Artikelgruppe.
<Description>	Beschreibung der Artikelgruppe.
<GroupNo>	Nummer der Artikelgruppe
<Addable>	True für Vorgabewert Hinzufügbarer Artikel
<Bonuspoints>	Vorgabewert Bonuspunkte
<CalculateOff>	Vorgabewert „Berechnen ab“
<Deposit>	Vorgabewert Pfandanteil
<HasDiscount>	Vorgabewert „Rabattfähig“
<IconIndex>	Index für Symbol
<IDValue>	Eindeutiger interner Zähler
<Image>	Bilddaten
<MajorArticle>	Vorgabewert Hauptartikel / Ergänzungsartikel
<ParentGroupRef>	Referenz auf übergeordnete Artikelgruppe
<SortOrder>	Sortierreihenfolge
<WarekindsRef>	Referenz auf Warengruppe
<Dependencies>	Siehe nächste Tabelle

Beschreibung für <Dependencies>:

Name:	Beschreibung:
<DependType>	Typ der Abhängigkeit. Mögliche Werte: adtAddArticle, adtAddGroup
<DependID>	Eindeutige ID.
<MustAdd>	True für Zwangsartikel.
<ForFree>	Artikel ist gratis.
<ArticleCount>	Anzahl der hinzuzufügenden Artikel

Beschreibung für <ArticleData>:

Name:	Beschreibung:
<ArticleRef>	Eindeutige ID des Artikels.
<ArticleNo>	Artikelnummer.
<ArticleGroupRef>	Referenz ID auf Artikelgruppe.
<Caption>	Artikelname
<Description>	Artikelbeschreibung
<Description2>	Zusätzliche Artikelbeschreibung
<Dependencies>	Zusatzartikel. Siehe oben.
<DefaultPrice>	Standardpreis
<BonusPoints>	Bonuspunkte
<CalculateOff>	„Berechnen ab“
<ContentMaterials>	Inhaltsstoffe
<Deposit>	Pfandanteil
<EANBarcode>	EAN Barcodenummer
<HasDiscount>	Rabattfähig
<MajorArticle>	True für Hauptartikel, False für Ergänzungsartikel
<Prices>	Zusätzliche Preise. Siehe nächste Tabelle
<SortOrder>	Sortierreihenfolge
<TempUnavailable>	Temporär ausverkauft
<Warekindsref>	ID auf Warengruppe

## E-Mail-Schnittstelle

Einleitung:

Die einfachste Implementierung auf Serverseite ist das Versenden der XML/JSON-Bestellung per E-Mail. WinOrder besitzt einen eingebauten E-Mail-Client, der in einstellbaren Abständen auf neue EShop-Bestellungen prüft.

Voraussetzung ist ein POP3/IMAP4 kompatibler E-Mail-Account. SSL-/TLS-Authentifizierung wie z.B. bei Google-Mail wird unterstützt.

Die XML-Daten müssen als Anhang hinzugefügt werden. Als Dateiendung wird „.xml“ oder „.json“ erwartet.

Sobald eine neue E-Mail-Bestellung abgerufen wurde lässt sich diese unter „Bestellungen/E-Mail Bestellungen“ anzeigen. WinOrder 6 unterstützt dabei auch mehrere E-Mail-Konten.

Wird ein gültiger XML/JSON-Anhang erkannt, speichert WinOrder diesen Anhang im überwachten Verzeichnis „EShop\Incoming“ und der EShop-Benachrichtigung wird angezeigt.

## Webservice Schnittstelle (REST-API)

Einleitung:

Die WinOrder REST-API ist eine einfache Schnittstelle zum Webshop mit Standard-Techniken HTTP(S)/REST/JSON. Der URL-Endpunkt sowie Benutzername/Kennwort wird in den Online-Shop Einstellungen vorgenommen:

Einstellungen für "My fancy REST-API shop"	Wert
Online-Shop aktiv	Ja
Online-Shop Name	My fancy REST-API shop
Übertragungsart	WinOrder-Webservice (REST)
Doppelbestellungen ignorieren (gleiche ID)	Nein
Kommentar in Bestellung einfügen	Ja
Zweitreechner-Benachrichtigung	<Keine>
Unbekannte Artikel	Immer abfragen
Artikelzuordnung	nach Artikelname
Aktion bei eindeutiger Bestellung	In Online-Bestellungen bleiben
Unbekannte Kunden automatisch anlegen	Ja
Zahlungsart abfragen	Nicht abfragen
Lieferzeit abfragen (Online-Tracking)	Nicht abfragen
Vorgabewert Fahrzeit (Online-Tracking)	20
Trinkgeld an Mitarbeiter auszahlen	Nein
Online-Shop Startseite	https://www.Ihre_Online-Shop_Startseite_.com
Webservice URL	https://my-online-shop.com/my-endpoint
Kunden-Nr./Benutzername/App-Key	MyUsername
Kennwort/App-Secret	****
Abfrageintervall in Min.	2

Ein einfaches PHP Beispiel für einen REST-API/JSON basierten Online-Shop Server finden Sie hier auf Github: <https://github.com/WinOrder/PHP-EShop-Server>

Verwenden Sie als Anbindung SSL mit Basic-Authentifizierung.  
Ihr Webserver muss zwei Endpunkte implementieren:

/GetNewOrders (HTTP GET)

Als Rückgabe an WinOrder werden neue Bestellungen als WinOrder JSON (oder auch XML) erwartet.

/SendTrackingStatus (HTTP POST)

WinOrder sendet eine Rückmeldung über den aktuellen Status der Bestellung an den Online-Shop. Die Authentifizierungsdaten werden neben Basic-Authentifizierung zusätzlich im Header übertragen, „username“ und „password“. Die JSON-Daten werden im Content übertragen.



JSON-Parameter:

Name	Typ	Beschreibung
ordersid	string	Bestell-ID vom Shop
trackingstatus	string	aktueller Bestellstatus, siehe nächste Tabelle
message	string	Status-Meldung
deliver_minutes	Integer	Geschätzte Lieferzeit in Minuten
deliver_eta	DateTime	Geschätzte Lieferuhrzeit
reject_reason	string	Ablehnungsgrund (für Status 7, 10)

Status:

TrackingStatus:	Beschreibung:
„0“ oder „OK“	Bestellung wurde erfolgreich empfangen und vom XML-Parser eingelesen. Meldung über neue EShop-Bestellung wird angezeigt.
„1“	Bestellung wurde über den EShop-Moderator Verarbeitet, die Bestellung wird gerade angenommen und in eine WinOrder Bestellung überführt.
„2“	Bestellung wurde gespeichert und ist jetzt in Zubereitung
„3“	Status „Küche“, wird derzeit noch nicht gesetzt
„4“	Status „Ofen“, wird derzeit noch nicht gesetzt
„5“	Bestellung wurde vom Fahrer gebucht (zugeordnet) und ist in Auslieferung
„6“	Bestellung wurde geliefert und ist abgeschlossen
„7“	Bestellung wurde im EShop-Moderator abgelehnt und gelöscht
„8“	Bestellung wurde rückerstattet
„9“	Annahme der Bestellung ist fehlgeschlagen (Fehler)
„10“	Bestellung wurde manuell storniert
„11“	Anderer Grund (nicht angegeben)

Ablauf:

WinOrder sendet /GetNewOrders im vorgegebenen Intervall (z.B. alle 2 Minuten) an den Webserver (Abfrage nach neuen Bestellungen). Liegt eine neue Bestellung vor wird diese als WinOrder JSON oder XML zurückgegeben (Content-Type „application/json“ oder „application/xml“).

WinOrder wertet die Bestelldaten aus und sendet /SendTrackingStatus „0“ = erfolgreich empfangen oder „9“= Fehler zurück. Bei erfolgreicher Übertragung sollte Ihre Webserver-Anwendung die Bestellung als verarbeitet markieren und nicht erneut anbieten.

Falls der Bestellstatus „0“ „erfolgreich durch WinOrder empfangen“ aus technischen Gründen auf Ihrem Webserver nicht gesetzt werden kann, besteht die Möglichkeit in den Online-Shop Einstellungen die mehrfach abgerufenen Bestellungen zu ignorieren. Diese Einstellung sollte mit Vorsicht gewählt werden, da dadurch Bestellungen auch „verschluckt“ werden könnten:

Bei Übernahme und Speichern der Bestellung wird Status=“2“ von WinOrder zurückgesendet usw.

Ist der Lieferzeit-Dialog aktiv wird zusätzlich die geschätzte Lieferzeit übertragen.

## Webservice Schnittstelle (SOAP)

Einleitung:

Durch die Verbreitung von DSL mit Flatrate besitzen mittlerweile die meisten Filialen einen Internetanschluss. Gefordert ist eine einfache Schnittstelle zum Webshop ohne aufwendige Einrichtung und Konfiguration. Außerdem soll HTTP als Standard-Protokoll verwendet werden, damit eine evt. Firewall-Konfiguration entfällt.

WinOrder enthält dazu einen SOAP 1.1 (Simple Object Access Protocol) Client, der den Online-Shop in einstellbaren Zeitabständen abfragt und Meldungen bzw. Tracking-Status zurücksendet.

Anforderungen auf Serverseite (Webshop):

Der Webshop muss einen SOAP 1.1 Server mit zunächst zwei Funktionen bereitstellen. Die Schnittstellen-Definition ergibt sich aus der WSDL Beschreibungsdatei die ebenfalls auf dem Server veröffentlicht wird.

Mit der Funktion „GetNewOrders“ prüft WinOrder in einstellbaren Zeitabständen auf neue Bestellungen und ruft diese ab:

```
string GetNewOrders(string Username, string Password)
```

WinOrder meldet sich mit Benutzernamen und Passwort am Webshop an und bekommt als Rückgabewert die XML-Bestelldaten. Das XML-Format ist identisch zum vorher beschriebenen

Format. Die XML-Daten werden von WinOrder eingelesen und eine Rückmeldung an den Server gesendet. Dazu ruft WinOrder die zweite Methode „SendTrackingStatus“ auf:

```
boolean SendTrackingStatus(string Username, string Password, string OrdersID, string TrackingStatus)
```

WinOrder sendet Benutzernamen und Passwort zur Authentifizierung sowie die eindeutigen Bestell-ID des Webshops und den Tracking-Status:

TrackingStatus:	Beschreibung:
„0“ oder „OK“	Bestellung wurde erfolgreich empfangen und vom XML-Parser eingelesen. Meldung über neue EShop-Bestellung wird angezeigt.
„1“	Bestellung wurde über den EShop-Moderator verarbeitet, die Bestellung wird gerade angenommen und in eine WinOrder Bestellung überführt.
„2“	Bestellung wurde gespeichert und ist jetzt in Zubereitung
„3“	Status „Küche“, wird derzeit noch nicht gesetzt
„4“	Status „Ofen“, wird derzeit noch nicht gesetzt
„5“	Bestellung wurde vom Fahrer gebucht und ist in Auslieferung

„6“	Bestellung wurde geliefert und ist abgeschlossen
„7“	Bestellung wurde im EShop-Moderator abgelehnt und gelöscht
„8“	Bestellung wurde rückerstattet
„9“	Annahme der Bestellung ist fehlgeschlagen
„10“	Bestellung wurde manuell storniert
„11“	Anderer Grund (nicht angegeben)

## WSDL Definition der Schnittstelle

```

<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" name="IWinOrderEshopservice"
targetNamespace="http://www.winorder.de" xmlns:tns="http://www.winorder.de"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/">
  <message name="GetNewOrders0Request">
    <part name="Username" type="xs:string"/>
    <part name="Password" type="xs:string"/>
  </message>
  <message name="GetNewOrders0Response">
    <part name="return" type="xs:string"/>
  </message>
  <message name="SendTrackingStatus1Request">
    <part name="Username" type="xs:string"/>
    <part name="Password" type="xs:string"/>
    <part name="OrdersID" type="xs:string"/>
    <part name="TrackingStatus" type="xs:string"/>
  </message>
  <message name="SendTrackingStatus1Response">
    <part name="return" type="xs:boolean"/>
  </message>
  <portType name="IWinOrderEshop">
    <operation name="GetNewOrders">
      <input message="tns:GetNewOrders0Request"/>
      <output message="tns:GetNewOrders0Response"/>
    </operation>
    <operation name="SendTrackingStatus">
      <input message="tns:SendTrackingStatus1Request"/>
      <output message="tns:SendTrackingStatus1Response"/>
    </operation>
  </portType>
  <binding name="IWinOrderEshopbinding" type="tns:IWinOrderEshop">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetNewOrders">
      <soap:operation soapAction="urn:WinOrderEshopIntf-
IWinOrderEshop#GetNewOrders" style="rpc"/>
      <input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WinOrderEshopIntf-IWinOrderEshop"/>
      </input>
      <output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WinOrderEshopIntf-IWinOrderEshop"/>
      </output>
    </operation>
    <operation name="SendTrackingStatus">
      <soap:operation soapAction="urn:WinOrderEshopIntf-
IWinOrderEshop#SendTrackingStatus" style="rpc"/>
      <input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WinOrderEshopIntf-IWinOrderEshop"/>
      </input>
      <output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

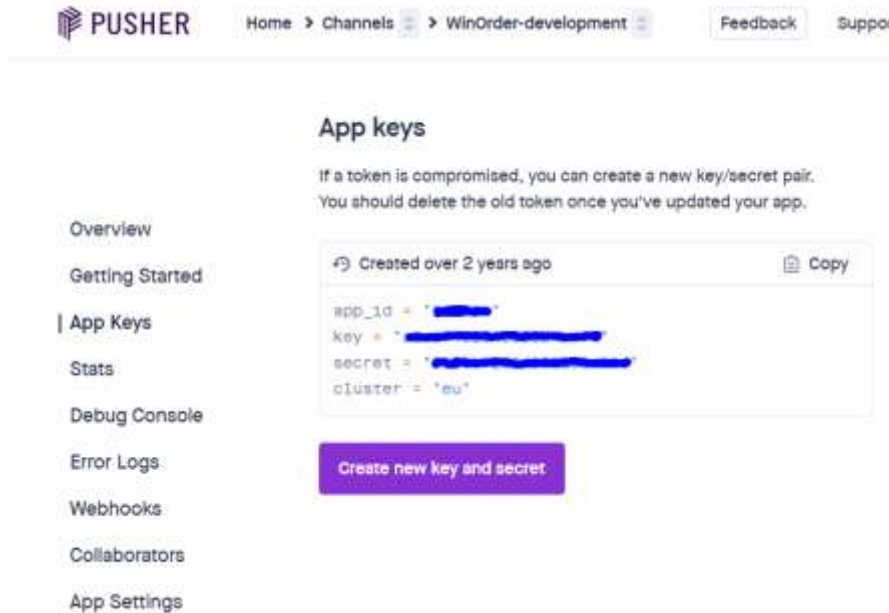
```

```
namespace="urn:WinOrderEShopIntf-IWinOrderEShop"/>
```

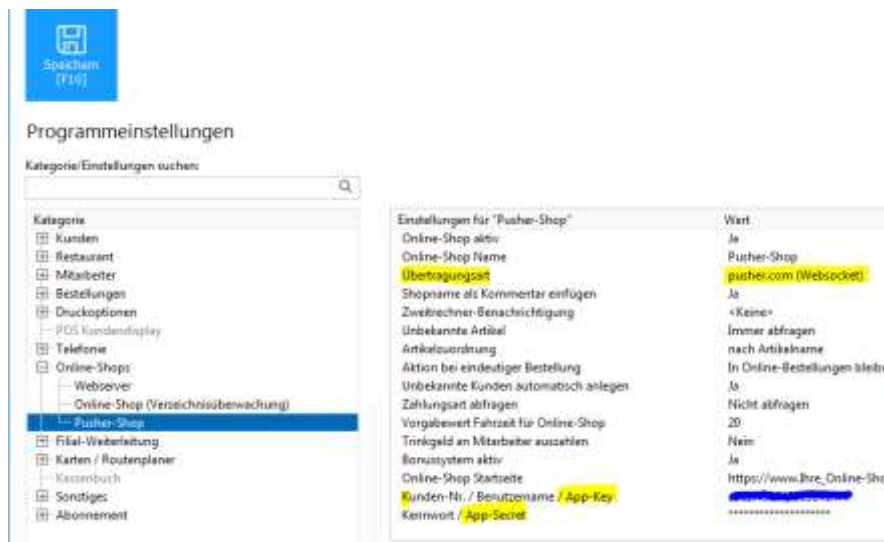
```
    </output>  
  </operation>  
</binding>  
<service name="IWinOrderEShopservice">  
  <port name="IWinOrderEShopPort" binding="tns:IWinOrderEShopbinding">  
    <soap:address  
location="http://localhost/Scripts/WinOrderEShop.dll/soap/IWinOrderEShop"/>  
  </port>  
</service>  
</definitions>
```

## Echtzeitübermittlung mit Pusher-Channels:

Mit [Pusher-Channels](#) können Bestellungen in Echtzeit an WinOrder übermittelt werden. Dazu lauscht WinOrder auf dem privaten Kanal „private-orders“ auf neue Bestellungen im JSON Format. Dazu wird im pusher.com eine neue App „WinOrder“ angelegt:



In WinOrder wird ein neuer Online-Shop mit der Übertragungsart „Pusher.com – (Websocket)“ hinzugefügt und App-Key/App-Secret eingetragen:



Bestellungen werden dann an den privaten Kanal „private-orders“ gepusht und in WinOrder in Echtzeit signalisiert.

## Artikelsynchronisierung über die SOAP Schnittstelle (WinOrder 9.0)

Datenkonflikte zwischen Internetshop und WinOrder können derzeit vom Anwender im EShop-Moderator aufgelöst werden. So können beispielsweise unbekannte Artikel aus dem Internetshop per Knopfdruck in WinOrder übertragen werden.

Eine Erweiterung der Webservice Schnittstelle ermöglicht die Synchronisierung der Artikeldaten per Knopfdruck. Die Artikelsynchronisierung wird vom Anwender über Stammdaten/Artikeleingabe, erweitertes (fett unterstrichenes) Menü „Artikel-/gruppen“ / Artikelsynchronisierung gestartet.

Artikeldaten aus dem Onlineshop werden automatisch in WinOrder importiert oder aktualisiert. Neue oder geänderte Artikeldaten können an den Onlineshop gesendet werden. Dazu muss diese Methode auf dem SOAP-Server implementiert werden:

```
String SynchArticle(string AUsername, string APassword, string ArticleXML): string;
```

WinOrder sendet zusammen mit Benutzernamen/Kennwort die Artikelstammdaten als XML. Als Rückgabewert von Ihrem Online-Shop werden die neuen/geänderten Artikeldaten im XML-, JSON- oder CSV-Format erwartet. WinOrder erkennt das Format automatisch.

## Filialweiterleitung / CallCenter Funktion

Ab WinOrder V4.0 Professional kann eine Bestellung als EShop-Bestellung an eine weitere Filiale gesendet werden. Als Übertragungsformat wird das hier beschriebene EShop-XML verwendet und auf Zielseite läuft die Bestellaufnahme über das überwachte EShop-Eingangsverzeichnis und den EShop-Moderator.

Als Übertragungsart kann ein E-Mail-Postfach, HTTP oder ein VPN-Netzwerkpfad verwendet werden.

## Eingebauter Webserver

WinOrder besitzt einen eingebauten Webserver. Dieser horcht auch Port 8080 (einstellbar) auf eingehende Bestellungen. Aktivieren Sie den Webserver in „Optionen/Einstellungen/Online-Shops/Webserver“. Wenn der Webserver erfolgreich gestartet wurde erscheint ein entsprechendes Icon in der Statusleiste:



Folgende Befehle werden unterstützt:

<http://localhost/index.html> (GET)

Zeigt die WinOrder Startseite im Browser an

<http://localhost/sendorder> (POST)



Senden Sie über diese URL die Bestellung direkt an WinOrder. Zugelassene MIME-Typen: text/xml und application/xml für Bestellungen im XML-Format sowie application/json und text/json für Bestellungen im JSON-Format.

HTTP-Statuscode 200 für erfolgreiche Zustellung und Verarbeitung, beim JSON-Format zusätzlich im Content:

```
{ "status": "success",  
  "data": null,  
  "message": "OK" }
```

<http://localhost/AscoOrder> (POST mit Basic-Authentication)

Empfängt Bestellungen im Pizza.de Format. Für die Authentifizierung werden Benutzername und Kennwort der EShop-Einstellung verwendet.

<http://localhost/webscr> (GET/POST)

Empfängt Bestellungen kompatibel zur PayPal-Checkout-Schnittstelle. Dokumentation dazu finden Sie hier:

[https://www.paypal.com/cgi-bin/webscr?cmd=p/pdn/howto\\_checkout-outside](https://www.paypal.com/cgi-bin/webscr?cmd=p/pdn/howto_checkout-outside)

Mit dieser Schnittstelle können Sie Ihren vorhandenen OnlineShop mit PayPal Schnittstelle an WinOrder anbinden. Beispiele für Shops mit PayPal Checkout-Funktion sind SimpleCart(js) oder JCart.

Beispiel Anforderung aus Ihrem Shop:

[http://localhost/webscr?cmd=cart&rm=0&upload=1&business=brett%40winorder.de&currency\\_code=USD&charset=utf-8&tax\\_cart=79.42&handling\\_cart=27&item\\_name\\_1=Zebra&quantity\\_1=5&amount\\_1=187.95&item\\_number\\_1=1&on0\\_1=thumb&os0\\_1=http%3A%2F%2F192.168.1.112%2Fsimplecart-js-2.2.3%2Ftest%2Ftest.png&on1\\_1=size&os1\\_1=Small&option\\_index\\_1=2&item\\_name\\_2=Cow&quantity\\_2=1&amount\\_2=18&item\\_number\\_2=2&on0\\_2=thumb&os0\\_2=http%3A%2F%2F192.168.1.112%2Fsimplecart-js-2.2.3%2Ftest%2Ftest.png&on1\\_2=size&os1\\_2=Medium&option\\_index\\_2=2&item\\_name\\_3=ba by+lion&quantity\\_3=1&amount\\_3=34.95&item\\_number\\_3=3&on0\\_3=size&os0\\_3=tiny&on1\\_3=thumb&os1\\_3=e.png&option\\_index\\_3=2&return=index.html](http://localhost/webscr?cmd=cart&rm=0&upload=1&business=brett%40winorder.de&currency_code=USD&charset=utf-8&tax_cart=79.42&handling_cart=27&item_name_1=Zebra&quantity_1=5&amount_1=187.95&item_number_1=1&on0_1=thumb&os0_1=http%3A%2F%2F192.168.1.112%2Fsimplecart-js-2.2.3%2Ftest%2Ftest.png&on1_1=size&os1_1=Small&option_index_1=2&item_name_2=Cow&quantity_2=1&amount_2=18&item_number_2=2&on0_2=thumb&os0_2=http%3A%2F%2F192.168.1.112%2Fsimplecart-js-2.2.3%2Ftest%2Ftest.png&on1_2=size&os1_2=Medium&option_index_2=2&item_name_3=ba by+lion&quantity_3=1&amount_3=34.95&item_number_3=3&on0_3=size&os0_3=tiny&on1_3=thumb&os1_3=e.png&option_index_3=2&return=index.html)

## **Beispielimplementierung in PHP**

Ein einfaches PHP Beispiel für einen REST-API/JSON basierten Online-Shop Server finden Sie hier auf Github:

<https://github.com/WinOrder/PHP-EShop-Server>