

- 1. Beschreibung..... 2
  - 1.1. Auswahl eines Artikels..... 6
  - 1.2. Anzeige eines zuvor ausgewählten Artikels..... 10
  - 1.3. Kopieren von Artikeln..... 12
  - 1.4. Preisabfrage von Artikeln..... 14
  - 1.5. Übergabe von Artikeln an den Warenkorb..... 16
  - 1.6 Testprogramm ..... 19
  - 1.7. Cookies..... 21
  - 1.8. Übertragung der Kommission in einer ZUGFeRD-Rechnung ..... 22
- 2. Anhang ..... 23
  - 2.0. Änderungshistorie ..... 23

## 1. Beschreibung

Dieses Dokument beschreibt die Anbindung von if-to-shop Anwendungen an bestehende Web-Shops.

Ziel ist es, dass aus den Anwendungen heraus der Benutzer im jeweiligen Shop Artikel auswählen und in die if-to-shop Anwendung übernehmen kann. Zu einem späteren Zeitpunkt kann der Artikel wieder aufgerufen werden. Im Hintergrund soll es möglich sein, die Preise für ausgewählte Artikel zu aktualisieren.

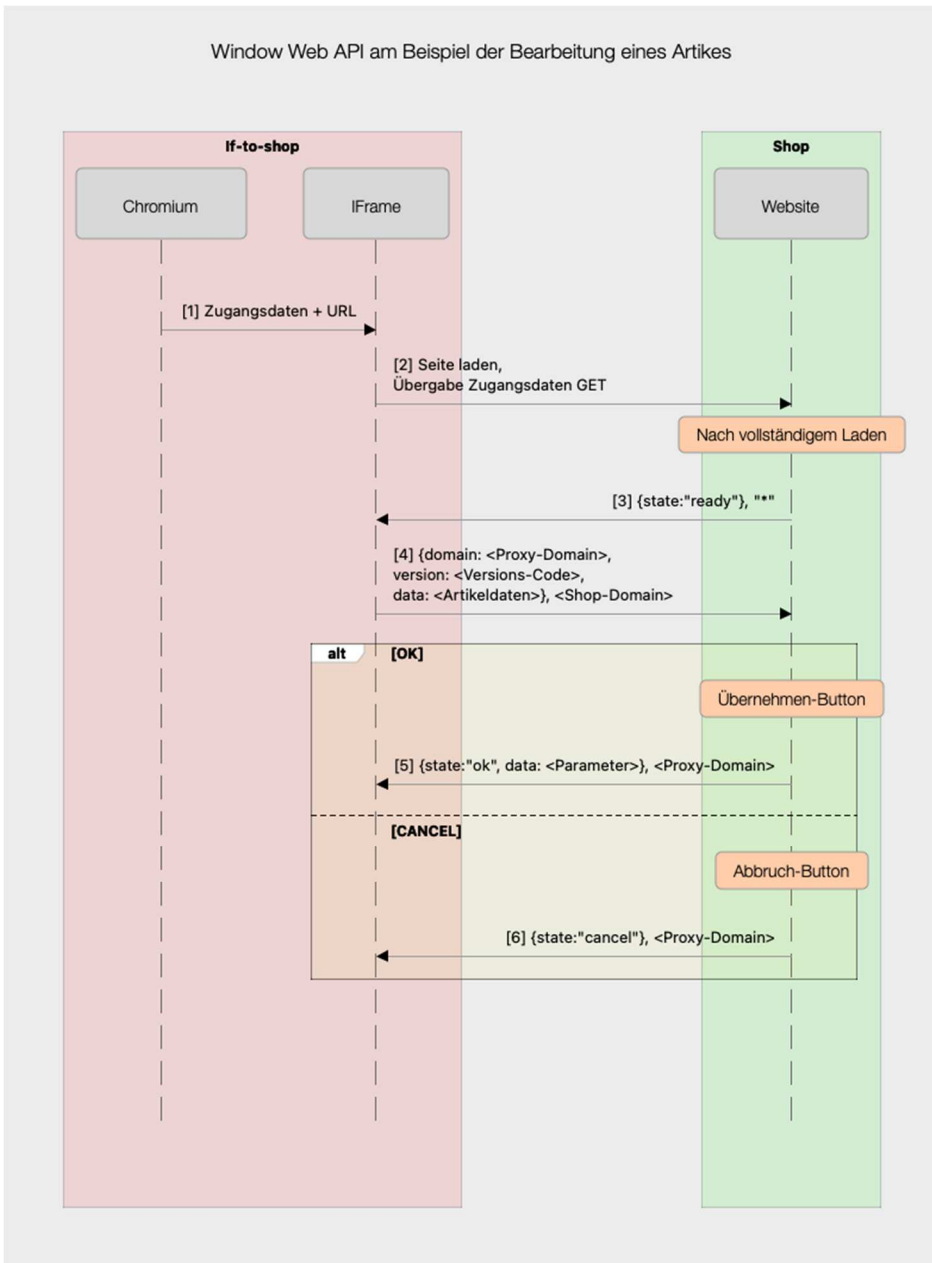
Abschließend soll der Benutzer ausgewählte Artikel in den Warenkorb des Shops legen und bestellen können.

Daraus ergeben sich fünf Szenarien

1. Auswahl eines Artikel, ggf. parametrisiert
2. Aufruf eines ausgewählten Artikels, wobei ggf. gewählte Parameter wieder gesetzt sein müssen
3. Erstellung einer Artikel-Kopie
4. Aktualisierung der Preisinformation
5. Übergabe einer Liste von Artikeln, welche im Shop in den Warenkorb gelegt und sofort bestellt werden können.

Die Kommunikation mit dem Shopsystem findet über das Window Web API statt.

Hierbei werden Informationen zwischen Domains mit unterschiedlichem Ursprung ausgetauscht.



Der Shop wird in einen IFrame eingebettet. Hierfür wird vom Shop-Betreiber je Szenario eine entsprechende URL zur Verfügung gestellt.

Diese URL wird von den if-to-shop Anwendungen mit den drei Parametern `username`, `customerNumber` und `password` über ein GET Request aufgerufen, um den Benutzer zu identifizieren. Die Kundennummer `customerNumber` ist hierbei optional.

Folgende Einstellungen müssen in der .htaccess Datei des WebServers vorgenommen werden:

- Die X-Frame-Optionen für die Einbettung ins iFrame
- Das Setzen der Cookies durch Drittanbieter

Beispiel:

```
<IfModule mod_headers.c>
  Header append X-Frame-Options SAMEORIGIN
</IfModule>

<IfModule mod_headers.c>
  Header always append X-Frame-Options "ALLOW-FROM https://shopproxy.p-s-s.de"
  Header set Content-Security-Policy "frame-ancestor 'self' https://shopproxy.p-s-s.de;"
  Header always edit Set-Cookie (.*) "$1; SameSite=none"
  Header always edit Set-Cookie (.*) "$1; Secure"
</IfModule>
```

Nachdem der IFrame vollständig geladen wurde sendet er, mit Hilfe der Javascript API, eine Nachricht an seinen Parent mit dem Inhalt `'ready'` und dem Target `'*'`.

Beispiel: `window.parent.postMessage({state: "ready"}, "*");`

Der Parent sendet nach Erhalt an den IFrame sein eigenes Target, welches bei zukünftigen Nachrichten vom IFrame verwendet werden muss und den Versionscode des Webservices. Zusätzlich kann die Nachricht weitere Informationen enthalten, beispielsweise eine Referenz auf einen Artikel oder eine Liste von Artikeln.

Das Rückgabe-Objekt muss, im Falle von Status `'Ok'`, alle unten beschriebenen Properties enthalten, aber nicht alle müssen gefüllt sein. Werte, die keine Pflichtfelder sind, können NULL oder einen Leerstrings enthalten. Pflichtfelder sind durch einen Stern markiert.

Im Verlauf der Dokumentation ist von weiteren Parametern im JSON Format die Rede. Die Anzahl der Parameter kann vom Shop frei vergeben werden. Sie werden in Form eines Arrays mit JSON Objekten übergeben und richten sich nach der Art des Artikels. Sie folgen immer der Form:

```
[
  {
    name*      : "Name des Parameters als eindeutige Kennung",
    title     : "Bezeichnung des Parameters für die Anzeige beim Benutzer",
    value*    : "Wert des Parameters",
    unit      : "Einheit des Parameters"
  },
  ...
]
```

Auch hier müssen alle Properties vorhanden sein, aber der Parameter `unit` kann NULL oder einen Leerstring enthalten.

Im Verlauf der Dokumentation ist von beschreibenden Parametern `descriptiveParameters` die Rede. Diese Parameter beschreiben den Artikel mit Werten, die in der aufrufenden Applikation sichtbar werden (im Gegensatz zu den weiteren Parametern – s.o. – die nur zur Parametrisierung des Artikels im WebShop dienen). Die beschreibenden Parameter sind optional und werden nicht an den WebShop zurückgegeben. Sie

folgen immer der Form:

```
[
  {
    label*   : "Bezeichnung des Parameters, z.B. Höhe, Länge, Width, Weight",
    value*   : "Wert des Parameters, z.B. 1.5",
    unit     : "Einheit des Parameters, z.B. qm, lfm"
  },
  ...
]
```

Bei der Übergabe von Artikeln an den Shop unterscheidet man drei Artikeltypen, die sich in der Behandlung der Artikel-Parameter unterscheiden:

### ***Standard-Artikel:***

Bei einem Standard-Artikel handelt es sich um einen Artikel, bei denen keine Artikel-Parameter im Shop ausgewählt werden können. Sind auswählbare Parameter vorhanden und wird für jede Variante eine eigene Artikelnummer generiert, so handelt es sich ebenfalls um einen Standard-Artikel.

Bei einem Standard-Artikel können Parameter angegeben werden, die jedoch nur intern vom Shop benutzt werden dürfen. Im Standard-Artikel muss die Property `parameterized` den Wert `false` enthalten.

Die Referenz und die Artikelnummer sind bei Standard-Artikel immer eindeutig.

### ***Parametrisierter Artikel mit anwenderseitiger Parameterverwaltung (Varianten-Artikel):***

Bei parametrisierten Artikeln sind im Shop Artikel-Parameter vorhanden, die vom Nutzer ausgewählt werden können. Hierbei wird jedoch nicht, wie beim Standard-Artikel, jedes mal eine neue Artikelnummer generiert, sondern die ausgewählten Parameter müssen als weitere Parameter aus dem Shop übergeben werden. Die weiteren Parameter müssen in der Property `parameters` eingetragen und die Property `parameterized` muss auf `true` gesetzt werden.

Weil die Referenz sowie die Artikelnummer bei parametrisierten Artikeln nicht eindeutig sind, wird über die Parameter ein Hashwert berechnet, um eine eindeutige Zuordnung zu erhalten. Der Hashwert wird von der IfToShop-Schnittstelle generiert und in der Property `parameterHash` eingetragen. Der generierte Hashwert wird an den Shop übergeben, von wo er, unverändert zurückgeliefert werden muss.

### ***Parametrisierter Artikel mit serverseitiger Parameterverwaltung:***

Bei parametrisierten Artikeln mit serverseitiger Parameterverwaltung werden die ausgewählten Parameter serverseitig gespeichert und müssen nicht über die Property `parameters` an die Schnittstelle übergeben werden. Die Referenz referenziert in diesem Fall nicht nur den Artikel, sondern ebenfalls die Parameter. Bei der serverseitigen Parameterverwaltung muss die Property `parameters` einen Leerstring bzw. null enthalten. Die Property `serverSidedParameters` kennzeichnet die serverseitige Speicherung der Parameter und muss den Wert `true` erhalten.

Die Referenz und die Artikelnummer sind bei parametrisierten Artikeln mit serverseitiger Parameterverwaltung immer eindeutig. Weil die Referenz sowohl den Artikel als auch die Parameter referenziert, muss, um einen Artikel zu kopieren, die Kopie serverseitig erfolgen (siehe Kapitel 1.3).

In den nachfolgenden Kapiteln werden jetzt alle möglichen Anwendungsfälle detailliert beschrieben.

## 1.1. Auswahl eines Artikels

### Ziel:

Der Benutzer kann im Shop surfen und sich die Artikel anschauen. Bei Interesse kann er einen Artikel auswählen, parametrisieren und ihn, anstatt in den Warenkorb zu legen, durch Klick auf eine Übernehmen-Schaltfläche in die if-to-shop Anwendung übernehmen. Hierfür soll die Schaltfläche im Shop „In den Warenkorb“ durch „In APP übernehmen“ ersetzt werden. Zusätzlich soll der Benutzer neben der „In APP übernehmen“-Schaltfläche auch eine Abbrechen-Schaltfläche sehen, um den Auswahl-Vorgang an dieser Stelle abbrechen zu können. Ein Abbruch kann auch durch einfaches Schließen des Fensters (z.B. über [X]) geschehen.

Bei Artikeln mit einer Verpackungseinheit ungleich 1 ist zu beachten, dass sich die Eigenschaft `pricePerUnit` auf die Einheit in der Eigenschaft `unit` bezieht. Bei einer Packung mit der Verpackungseinheit 100 Stück müsste der Gesamtnettopreis der Verpackung also durch 100 geteilt werden.

### Benötigt:

- URL des Shops
- Version der Schnittstelle

### Beenden des Dialogs durch:

- Abbruch-Schaltfläche
- Auswählen-Schaltfläche

### Message an den IFrame nach Erhalt der 'ready'-Message:

```
{
  domain      : 'Domain des IFrame-Parent',
  version     : 'aktueller Versionscode'
}
```

### Message vom IFrame im Falle des Abbruchs:

```
{
  state*      : 'cancel'
}
```

### Message vom IFrame im Falle der Auswahl:

```
{
  state*: 'ok',
  data*  : {
    reference*      : 'Eindeutige Referenz auf den Artikel',
    title*          : 'Bezeichnung des Artikels',
    shortDescr*    : 'Kurztext des Artikels',
    number*        : 'Artikelnummer',
    pricePerUnit*  : 'Preis pro Einheit',
    currency*      : 'Währung',
    packagingUnit* : 'VE, Zahl, Dezimalzeichen Punkt, z.B. 5.34',
    unit*          : 'Einheit z.B. m',
    longDescr*     : 'Beschreibung',
    articleGroup   : 'Bezeichnung der Artikelgruppe',
    gtinNumber     : 'GTIN Nummer des Artikels',
    graphic*       : 'URL oder Base64 kodiert',
    serverSidedParameters : 'true|false - Eventuelle Parameter des Artikels
    werden auf dem Server verwaltet und nicht über die Schnittstelle übergeben.',
    parameterized  : 'true|false, default: false, Parametrisierter
    Artikel)
    parameters     : [ weitere Parameter im JSON Format ],
    descriptiveParameters : [ beschreibende Parameter im JSON Format ]
  }
}
```

```
}
}
```

### Fehlerbehandlung:

- Internetverbindung ist nicht vorhanden: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'Error offline'** zurückgeliefert
- Der Shop ist nicht erreichbar: Dialog muss, z.B. über x, geschlossen werden. Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'cancel'** zurückgeliefert
- Fehlerhafte Zugangsdaten: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern
- Wartungsarbeiten, Störungen etc.: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern

### Beispiele:

Beispiel Platte:

```
{
  state: 'ok',
  data: {
    reference      : '88439854',
    title          : '2SPA19weiss',
    shortDescr    : 'Dekorspanplatte 19 mm weiß',
    number         : 'DEK19W',
    pricePerUnit  : '8,50',
    currency       : 'EUR',
    packagingUnit : '5,80',
    unit           : 'm²',
    longDescr     : 'Dekorspanplatte 19 mm weiß formaldehydfrei, DIN 123',
    articleGroup  : 'Platten / Spanplatten / UNI / 19 mm',
    gtinNumber    : null
    graphic       : 'https://www.ultimate-shop24.de/graphics/88439854-1',
    serverSidedParameters: false,
    parameterized : false,
    descriptiveParameters: [
      {
        label: 'Länge',
        value: '2700',
        unit: 'mm'
      },{
        label: 'Breite',
        value: '2070',
        unit: 'mm'
      },{
        label: 'Stärke',
        value: '19',
        unit: 'mm'
      },{
        label: 'Farbe',
        value: 'weiß'
      }
    ]
  }
}
```

## Beispiel Topfband:

```
{
  state: 'ok',
  data: {
    reference      : '88445544',
    title          : 'Topfband 90 Grad',
    shortDescr     : 'Topfband 90 Grad, 35 mm Bohrung',
    number         : 'TFB-453',
    pricePerUnit   : '2,50',
    currency       : 'EUR',
    packagingUnit  : '50',
    unit           : 'Stück',
    longDescr      : 'Topfband in ALU mit einem Öffnungswinkel von 90 Grad',
    articleGroup   : 'Beschläge / Möbel / Topfband',
    gtinNumber     : null
    graphic        : 'https://www.ultimate-shop24.de/graphics/88445544-3',
    serverSiededParameters: false,
    parameterized  : false,
    descriptiveParameters: [
      {
        label: 'Material',
        value: 'ALU'
      },{
        label: 'Oberfläche',
        value: 'blank'
      },{
        label: 'Öffnungswinkel',
        value: '90',
        unit: 'Grad'
      },{
        label: 'Bohrung',
        value: '35',
        unit: 'mm'
      }
    ]
  }
}
```

## Beispiel Fenster:

```
{
  state: 'ok',
  data: {
    reference      : '88443547',
    title          : 'Kunststofffenster',
    shortDescr     : 'Kunststofffenster W705 weiss',
    number         : 'KSF-W705-W',
    pricePerUnit   : '250',
    currency       : 'EUR',
    packagingUnit  : '1',
    unit           : 'Stück',
    longDescr      : 'Kunststofffenster, weiss Dreh Br.=1000mm, Hö.=1200mm',
    articleGroup   : 'Kunststofffenster',
    gtinNumber     : null
    graphic        : 'https://www.ultimate-shop24.de/graphics/88443544-2',
    serverSiededParameters: false,
    parameterized  : true,
    parameters: [
      {
        name: 'width',
        title: 'Breite',
        value: '1000',
      }
    ]
  }
}
```



```

    unit: 'mm'
  }, {
    name: 'height',
    title: 'Höhe',
    value: '1200',
    unit: 'mm'
  }, {
    name: 'color',
    title: 'Farbe',
    value: 'weiss'
  }, {
    name: 'stop',
    title: 'Anschlag',
    value: 'Dreh'
  }
]
}
}

```

Beispiel Fenster mit serverseitigen Parametern:

```

{
  state: 'ok',
  data: {
    reference      : '1612869861',
    title          : 'Kunststofffenster',
    shortDescr    : 'Kunststofffenster Classicline ohne Folierung',
    number        : '1612869861',
    pricePerUnit  : '202.00',
    currency      : 'EUR',
    packagingUnit : '1',
    unit          : 'Stück',
    longDescr     : 'Kunststofffenster Classicline ohne Folierung\nDichtung grau\n
                    Br.=1000mm, Hö.=1000mm\nEntwässerung nach vorne\nDreh-Kipp links
                    Griff Classicline\n',
    articleGroup  : 'Kunststofffenster',
    gtinNumber    : null
    graphic       : 'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAdYAAAIICAYAAADTxs
                    QVR4XuydB1RUydLH/2bXsOacs65pjZgTGBHMARazooiYdVGUIJMilnBHDBn15x1z',
    serverSidedParameters: true,
    parameterized  : false,
    parameters    : null
    descriptiveParameters: null
  }
}

```

## 1.2. Anzeige eines zuvor ausgewählten Artikels

### Ziel:

Der Benutzer kann einen zuvor ausgewählten Artikel wieder aufrufen, um ihn zu sichten oder ggf. erweiterte Parameter verändern zu können.

### Benötigt:

- URL des Shops
- Version der Schnittstelle
- Referenz des Artikels und ggf. weitere Parameter

### Beenden des Dialogs durch:

- Abbruch-Schaltfläche
- Auswählen-Schaltfläche

### Message an den IFrame nach Erhalt der 'ready'-Message:

```
{
  domain      : 'Domain des IFrame-Parent',
  version     : 'aktueller Versionscode'
  data       : {
    reference  : 'Eindeutige Referenz auf den Artikel (Pflicht)',
    parameters : [ weitere Parameter, wie bei Auswahl übergeben ],
    ...
  }
}
```

### Fehlerbehandlung:

- Internetverbindung ist nicht vorhanden: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'Error offline'** zurückgeliefert
- Der Shop ist nicht erreichbar: Dialog muss, z.B. über x, geschlossen werden. Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'cancel'** zurückgeliefert
- Fehlerhafte Zugangsdaten: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern
- Wartungsarbeiten, Störungen etc.: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern

### Beispiele:

Beispiel Datenübergabe Platte:

```
{
  ...
  data: {
    reference : '88439854',
    parameters : []
  }
}
```

Beispiel Datenübergabe Topfband:

```
{
  ...
  data: {
    reference : '88445544',
    parameters : []
  }
}
```

Beispiel Datenübergabe Fenster:

```
{
  ...
  data: {
    reference: '88443547',
    parameters: [
      {name: 'width', title: 'Breite', value: '1000', unit: 'mm'},
      {name: 'height', title: 'Höhe', value: '1200', unit: 'mm'},
      {name: 'color', title: 'Farbe', value: 'weiss'},
      {name: 'stop', title: 'Anschlag', value: 'Dreh'},
      {name: 'profile', title: 'Profil', value: 'W705'},
    ]
  }
}
```

**Message vom IFrame im Falle des Abbruchs:**

```
{
  state*      : 'cancel'
}
```

**Message vom IFrame im Falle der Auswahl:**

```
{
  state*: 'ok',
  data*  : {
    reference*      : 'Eindeutige Referenz auf den Artikel',
    title*          : 'Bezeichnung des Artikels',
    shortDescr*    : 'Kurztext des Artikels',
    number*        : 'Artikelnummer',
    pricePerUnit*  : 'Preis pro Einheit',
    currency*      : 'Währung',
    packagingUnit* : 'VE, Zahl, Dezimalzeichen Punkt, z.B. 5.34',
    unit*          : 'Einheit z.B. m',
    longDescr*     : 'Beschreibung',
    articleGroup   : 'Bezeichnung der Artikelgruppe',
    gtinNumber     : 'GTIN Nummer des Artikels',
    graphic*       : 'URL oder Base64 kodiert',
    serverSidedParameters : 'true|false - Eventuelle Parameter des Artikels
    werden auf dem Server verwaltet und nicht über die Schnittstelle übergeben.',
    parameterized  : 'true|false, default: false, Parametrisierter
    Artikel)
    parameters     : [ weitere Parameter im JSON Format ],
    descriptiveParameters : [ beschreibende Parameter im JSON Format ]
  }
}
```

Beispiele:

An dieser Stelle finden sich keine Beispiele, weil sie mit denen der Auswahl eines Artikels identisch sind.

### 1.3. Kopieren von Artikeln

#### Ziel:

Es kann vorkommen, dass die Parameter eines Artikels nicht an die Schnittstelle übergeben werden, sondern serverseitig gespeichert werden. In diesen Fällen referenziert der Parameter [reference](#) nicht nur den Artikel, sondern auch seine Parameter.

Um dieses Server-Verhalten zu übermitteln, muss bei der Auswahl und beim Ändern des Artikels die Einstellung [serverSiededParameters](#) auf true gesetzt werden.

Soll ein solcher Artikel später kopiert und überarbeitet werden, muss diese Kopie serverseitig erfolgen, damit beim Ändern der Parameter nicht der Originalartikel überschrieben wird. Zur serverseitigen Herstellung einer Kopie und Rückgabe der neuen Referenz dient dieser Aufruf, bei welchem mehrere Artikel auf einmal kopiert werden können. Diese Kommunikation verläuft ohne Benutzerinteraktivität. Daher wird keine Kommunikation zwischen IFrame und Parent benötigt. Stattdessen erfolgt der Aufruf direkt über ein POST Request an den Shop.

#### Benötigt:

- URL des Shops

#### POST Parameter:

Name: 'request', Value: (JSON encoded):

```
{
  username      : 'Benutzername',
  customerNumber : 'Kundennummer',
  password      : 'Passwort',
  articles      : [
    'Eindeutige Referenz auf den ersten Artikel',
    'Eindeutige Referenz auf den zweiten Artikel',
    ...
  ]
}
```

#### Fehlerbehandlung:

- Internetverbindung ist nicht vorhanden: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein ['Error offline'](#) zurückgeliefert
- Der Shop ist nicht erreichbar: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein ['cancel'](#) zurückgeliefert
- Fehlerhafte Zugangsdaten: Der Web-Shop muss, bei fehlerhaften Zugangsdaten, ein ['cancel'](#) zurückliefern
- Wartungsarbeiten, Störungen etc.: Der Web-Shop muss, bei Wartungsarbeiten etc., ein ['cancel'](#) zurückliefern

#### Beispiel:

```
{
  ...
  articles: ['88439854', '88445544']
}
```

#### Die Rückgabe:

Die Rückgabe erfolgt im JSON Format:

```
[
  {
    reference*      : 'Eindeutige Referenz auf den ersten Artikel',
```

```
valid*           : true|false (Artikelverfügbarkeit),
referenceCopy*   : 'Eindeutige Referenz auf die Kopie des ersten Artikels,
                  wenn valide'
}, {
reference*       : 'Eindeutige Referenz auf den zweiten Artikel',
valid*          : true|false (Artikelverfügbarkeit),
referenceCopy*   : 'Eindeutige Referenz auf die Kopie des zweiten Artikels,
                  wenn valide'
},
...
]
```

**Beispiel:**

```
[
{
reference       : '88439854',
valid          : true,
referenceCopy   : '88489855'
}, {
reference       : '88445544',
valid          : false,
referenceCopy   : null
}
]
```

## 1.4. Preisabfrage von Artikeln

### Ziel:

Für die übergebenen Artikel-Referenzen soll der aktuelle Preis zurückgegeben werden. Diese Kommunikation verläuft ohne Benutzerinteraktivität. Daher wird keine Kommunikation zwischen IFrame und Parent benötigt. Stattdessen erfolgt der Aufruf direkt über ein POST Request an den Shop.

Damit beim POST Request die Daten korrekt im Body übermittelt werden, muss als Content-Type `,application/x-www-form-urlencoded'` angegeben werden.

Falls der übergebene Artikel Parameter enthält, muss die Property parameterHash mit an den Shop übergeben und vom Shop unverändert zurückgegeben werden. Sind keine Parameter enthalten, muss parameterHash null oder einen Leerstring enthalten.

Bei Artikeln mit einer Verpackungseinheit ungleich 1 ist zu beachten, dass sich die Eigenschaft `pricePerUnit` in der Rückgabe auf die Einheit in der Eigenschaft `unit` bezieht. Bei einer Packung mit der Verpackungseinheit 100 Stück müsste der Gesamtnettopreis der Verpackung also durch 100 geteilt werden.

### Benötigt:

- URL des Shops

### POST Parameter:

Name: 'request', Value: (JSON encoded):

```
{
  username      : 'Benutzername',
  customerNumber : 'Kundennummer',
  password      : 'Passwort',
  articles      : [
    {
      reference    : 'Eindeutige Referenz auf den Artikel',
      parameterHash : 'Ein Hash-Wert über Parameter, welcher vom Shop zurückgegeben
                        werden muss.',
      parameters   : []
    },
    ...
  ]
}
```

### Fehlerbehandlung:

- Internetverbindung ist nicht vorhanden: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein `'Error offline'` zurückgeliefert
- Der Shop ist nicht erreichbar: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein `'cancel'` zurückgeliefert
- Fehlerhafte Zugangsdaten: Der Web-Shop muss, bei fehlerhaften Zugangsdaten, ein `'cancel'` zurückliefern
- Wartungsarbeiten, Störungen etc.: Der Web-Shop muss, bei Wartungsarbeiten etc., ein `'cancel'` zurückliefern

### Beispiel für alle drei zuvor ausgewählten Artikel:

```
{
  ...
  articles: [
    {
      reference    : '88439854',
      parameterHash : '',
      parameters   : []
    }, {
      reference    : '88445544',
      parameterHash : '',
      parameters   : []
    }
  ]
}
```

```

    }, {
      reference      : '88443547',
      parameterHash : '786345',
      parameters    : [
        {name: 'width', title: 'Breite', value: '1000', unit: 'mm'},
        {name: 'height', title: 'Höhe', value: '1200', unit: 'mm'},
        {name: 'color', title: 'Farbe', value: 'weiss'},
        {name: 'stop', title: 'Anschlag', value: 'Dreh'},
        {name: 'profile', title: 'Profil', value: 'W705'}
      ]
    }
  ]
}

```

**Die Rückgabe erfolgt ebenfalls im JSON Format (alles Pflichtfelder):**

```

[
  {
    reference*      : 'Eindeutige Referenz auf den Artikel',
    parameterHash* : 'Der Hash-Wert über Parameter, welcher vom Shop wieder
                    zurückgegeben wird.',
    valid*          : true|false (Artikelverfügbarkeit),
    pricePerUnit*  : 'Preis pro Einheit',
    currency*       : 'Währung'
  },
  ...
]

```

**Beispiel für alle drei zuvor ausgewählten Artikel:**

```

[
  {
    reference      : '88439854',
    parameterHash  : '',
    valid          : true,
    pricePerUnit  : '8.45',
    currency       : 'EUR'
  }, {
    reference      : '88445544',
    parameterHash  : '',
    valid          : true,
    pricePerUnit  : '2.50',
    currency       : 'EUR'
  }, {
    reference      : '88443547',
    parameterHash  : '883476',
    valid          : true,
    pricePerUnit  : '250.00',
    currency       : 'EUR'
  }
]

```

## 1.5. Übergabe von Artikeln an den Warenkorb

### Ziel:

Die übergebenen Artikel mit ggf. vorhandenen Parametern sollen in den Warenkorb gelegt und dieser aufgerufen werden. Aus dem Warenkorb heraus soll dann direkt die Bestellung erfolgen können. Im Warenkorb darf der Anwender Artikel löschen oder die Menge der zu bestellenden Artikel ändern. **Die Parameter eines Artikels dürfen nicht verändert werden**, weitere Artikel dürfen dem Warenkorb jedoch hinzugefügt werden.

Falls der übergebene Artikel Parameter enthält, muss die Property `parameterHash` mit an den Shop übergeben und vom Shop unverändert zurückgegeben werden. Sind keine Parameter enthalten, muss `parameterHash` null oder einen Leerstring enthalten.

Bei Artikeln mit einer Verpackungseinheit ungleich 1 ist zu beachten, dass sich die Eigenschaft `count` auf die Menge der Einheiten bezieht und nicht auf die Menge der Verpackungseinheiten. Werden zum Beispiel 250 Schrauben mit einer Verpackungseinheit von 100 Stück bestellt, so wird in der Eigenschaft `count` 250 an den Shop übergeben. Weil die if-to-shop-Anwendung die möglichen Verpackungsgrößen im Shop nicht kennt, muss der Shop die nächste passende Größe ermitteln und als Menge im Warenkorb eintragen. Im obigen Fall würde der Shop die Menge auf 300 Stück erhöhen und im Warenkorb 3 VE \* 100 Stück eintragen.

In der Rückgabe muss die tatsächlich bestellte Menge von 300 Stück sowie der Gesamtnettopreis der Bestellung eingetragen werden.

Ein ggf. übergebene Kommissionsnummer und Lieferdatum soll auf der Rechnung beim Artikel mit aufgeführt werden. Nähere Angaben hierzu entnehmen Sie bitte Kapitel 1.8.

### Benötigt:

- URL des Shops

### Beenden des Dialogs durch:

- Abbruch-Schaltfläche
- Auswählen-Schaltfläche

### Message an den IFrame nach Erhalt der 'ready'-Message:

```
{
  domain      : 'Domain des IFrame-Parent',
  version     : 'aktueller Versionscode',
  data       : {
    username   : 'Benutzername',
    customerNumber : 'Kundennummer',
    password   : 'Passwort',
    data       : [
      {
        reference           : 'Referenz',
        parameterHash      : 'Ein Hash-Wert über Parameter, welcher vom Shop bei
                             Bestellung zurückgegeben werden muss.',
        count               : 'Menge',
        commissionNumber    : 'Angabe einer Kommissionsnummer für die Rechnung',
        deliveryDate        : 'Angabe eines Lieferdatums für den Artikel',
        parameters          : [], ...
      },
      ...
    ]
  }
}
```



**Fehlerbehandlung:**

- Internetverbindung ist nicht vorhanden: Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'Error offline'** zurückgeliefert
- Der Shop ist nicht erreichbar: Dialog muss, z.B. über x, geschlossen werden. Die aufrufende Anwendung bekommt, aus der Schnittstelle, ein **'cancel'** zurückgeliefert
- Fehlerhafte Zugangsdaten: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern
- Wartungsarbeiten, Störungen etc.: Der Web-Shop muss im Dialog eine Fehlermeldung anzeigen und, beim Schliessen des Dialogs, ein **'cancel'** zurückliefern

**Beispiel für alle drei zuvor ausgewählten Artikel:**

```

...
data: [
  {
    reference      : '88439854',
    parameterHash  : '',
    count          : 12,
    commissionNumber : null,
    deliveryDate   : null,
    parameters     : [],
    ...
  }, {
    reference      : '88445544',
    parameterHash  : '',
    count          : 3,
    commissionNumber : null,
    deliveryDate   : null,
    parameters     : [],
    ...
  }, {
    reference      : '88443547',
    parameterHash  : '554362',
    count          : 1,
    commissionNumber : null,
    deliveryDate   : null,
    parameters     : [
      {name: 'width',      title: 'Breite',      value: '1000', unit: 'mm'},
      {name: 'height',    title: 'Höhe',      value: '1200', unit: 'mm'},
      {name: 'color',     title: 'Farbe',      value: 'weiss'},
      {name: 'stop',      title: 'Anschlag',   value: 'Dreh'},
      {name: 'profile',   title: 'Profil',     value: 'W705'},
    ],
    ...
  }
]

```

**Message vom IFrame im Falle des Abbruchs:**

```

{
  state*: 'cancel'
}

```

**Message vom IFrame im Falle der Bestellung:**

```

{
  state*: 'ok',
  data*   : [
    {
      reference*      : 'Eindeutige Referenz auf den Artikel',
      parameterHash* : 'Hashwert der Parameter',
    }
  ]
}

```

```

    valid*           : true|false (Artikelverfügbarkeit),
    count*           : 'Anzahl im Warenkorb',
    price*           : 'Nettopreis für Artikel*Menge',
    currency*        : 'Währung',
    commissionNumber : 'Angabe einer Kommissionsnummer',
    deliveryDate     : 'Angabe eines Lieferdatums'
  },
  ...
]
}

```

### Beispiel für alle drei zuvor ausgewählten Artikel:

```

{
  state: 'ok',
  data: [
    {
      reference       : '88439854',
      parameterHash   : '',
      valid           : true,
      count           : 12,
      price           : '101.40',
      currency        : 'EUR',
      commissionNumber : null,
      deliveryDate    : null
    }, {
      reference       : '88445544',
      parameterHash   : '',
      valid           : true,
      count           : 3,
      price           : '7.50',
      currency        : 'EUR',
      commissionNumber : null,
      deliveryDate    : null
    }, {
      reference       : '88443547',
      parameterHash   : '943182',
      valid           : true,
      count           : 1,
      price           : '250.00',
      currency        : 'EUR',
      commissionNumber : null,
      deliveryDate    : null
    }
  ]
}

```

## 1.6 Testprogramm

### Download:

if-to-shop stellt ein Programm zum Testen der Schnittstelle unter Windows zur Verfügung. Unter folgender URL kann eine ZIP-Datei im 32 Bit-Format mit allen benötigten Dateien und die aktuelle Schnittstellenbeschreibung heruntergeladen werden:

<https://www.if-to-shop.com/download/ifToShop32.zip>

Um das Passwort zum Entpacken der ZIP-Datei zu erhalten, senden Sie bitte eine Mail an [info@if-to-shop.com](mailto:info@if-to-shop.com).

Nach dem Entpacken des heruntergeladenen Zip-Archivs wechseln Sie in das Unterverzeichnis shop. Erstellen Sie eine Kopie der darin befindlichen Datei [shops.json.example](#), benennen Sie diese Kopie um in [shops.json](#) und öffnen Sie die Datei in einem Texteditor.

```
[
  {
    "aesKey"      : "32Byte AES Key im Hexformat",
    "name"       : "NAME DES SHOPS",
    "username"   : "BENUTZERNAME",
    "customerNumber": "Kundennummer",
    "password"   : "PASSWORT",
    "domain"    : "https://www.mein-shop.de",
    "browse"    : "https://www.mein-shop.de/index-erp.php?action=browse
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%",
    "gtinBrowse" : "https://www.mein-shop.de/index-erp.php?action=browse
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%",
    "edit"      : "https://www.mein-shop.de/index-erp.php?action=edit
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%",
    "copy"     : "https://www.mein-shop.de/index-erp.php?action=copy
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%",
    "priceUpdate" : "https://www.mein-shop.de/index-erp.php?action=priceupdate
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%",
    "addToCart"  : "https://www.mein-shop.de/index-erp.php?action=addToCart
      &username=%USERNAME%&customerNumber=%CUSTOMER_NUMBER%&password=%PASSWORD%"
  }
]
```

### Die Beschreibung der Parameter im Einzelnen:

- **aesKey:** OPTIONAL Bei Angabe eines 32 Byte langen Schlüssels erfolgt die Übergabe der Zugangsdaten AES verschlüsselt im JSON Format. Nähere Informationen hierzu finden sich am Ende dieses Abschnitts.
- **name:** Hier geben Sie den Namen Ihres Shops an.
- **username:** Hier geben Sie den Benutzernamen des Kunden an.
- **customerNumber:** Hier können Sie zusätzlich die Kundennummer eingeben, falls dies für den Shop erforderlich ist.
- **password:** Hier geben Sie das Passwort des Kunden an.
- **domain:** Hier geben Sie die Domain Ihres Shops an.
- **browse:** Hier geben Sie die URL an, die benutzt werden soll, um nach Artikeln in Ihrem Shop zu suchen. Passen Sie die URL entsprechend Ihren Bedürfnissen an. Für die Platzhalter `%USERNAME%`, `%CUSTOMER_NUMBER%` und `%PASSWORD%` werden vom Programm dann die oben angegebenen Zugangsdaten eingesetzt.
- **edit:** Hier geben Sie die URL an, die benutzt werden soll, um einen Artikel in Ihrem Shop erneut aufzurufen. Passen Sie die URL entsprechend Ihren Bedürfnissen an. Für die Platzhalter `%USERNAME%`, `%CUSTOMER_NUMBER%` und `%PASSWORD%` werden vom Programm dann die oben angegebenen Zugangsdaten eingesetzt.
- **copy:** Hier geben Sie die URL an, die benutzt werden soll, um Artikel zu kopieren. Sollte Ihr Shop diese Funktion nicht unterstützen, weil die Parameter - falls vorhanden - an die Schnittstelle übergeben werden und nicht serverseitig gespeichert werden, lassen Sie diese URL einfach leer.

Ansonsten passen Sie die URL entsprechend Ihren Bedürfnissen an. Für die Platzhalter %%USERNAME%%, %%CUSTOMER\_NUMBER%% und %%PASSWORD%% werden vom Programm dann die oben angegebenen Zugangsdaten eingesetzt.

- priceUpdate: Hier geben Sie die URL an, die benutzt werden soll, um eine Preisanfrage für ausgewählte Artikel in Ihrem Shop durchzuführen. Passen Sie die URL entsprechend Ihren Bedürfnissen an. Für die Platzhalter %%USERNAME%%, %%CUSTOMER\_NUMBER%% und %%PASSWORD%% werden vom Programm dann die oben angegebenen Zugangsdaten eingesetzt.
- addToCart: Hier geben Sie die URL an, die benutzt werden soll, um ausgewählte Artikel in Ihrem Shop in den Warenkorb zu legen. Passen Sie die URL entsprechend Ihren Bedürfnissen an. Für die Platzhalter %%USERNAME%%, %%CUSTOMER\_NUMBER%% und %%PASSWORD%% werden vom Programm dann die oben angegebenen Zugangsdaten eingesetzt.

Anschließend können Sie das Testprogramm [IfToShop.exe](#) starten und Ihre Tests durchführen.

Für den Test der Schnittstelle werden weitere Dateien im dataTEST-Ordner angelegt und mit den Angaben aus der shops.json gefüllt. Hierbei wird unter [username](#) in der erstellten [Shop].json ‚error ExistsProvider‘ eingetragen, da der Anbieter in der Schnittstelle noch nicht freigeschaltet wurde. Die Angaben für den Verbindungsaufbau bezieht die Schnittstelle weiterhin aus der shops.json im Unterverzeichnis shop.

Die Buttons ‚Registrierung‘ und ‚Provider fehlt‘ haben für Ihre Tests keine Relevanz, sondern werden nur intern von PinnCalc genutzt.

#### **AES Verschlüsselung:**

Wenn Sie für in der [shops.json](#) als aesKey einen 32 Byte Hex-kodierten String (234ef284....) angeben, werden die Werte [username](#), [password](#), [customerNumber](#) und ein Zeitstempel im JSON Format AES verschlüsselt übergeben. Hierfür werden anstatt der oben angegebenen, die folgenden beiden Parameter in den URL benutzt: %%CREDENTIALS%% und %%IV%%. Die beiden Parameter werden im Base64 Format eingefügt. So könnte die Browse – URL z.B. so aussehen:

<https://www.mein-shop.de/index-erp.php?action=browse&credentials=%%CREDENTIALS%%&iv=%%IV%%>

Bei der Verschlüsselung wird ein Initialisierungsvektor generiert, welcher über den Parameter IV im Request übergeben wird. Auf Seite des Shops kann mit diesen Daten das JSON mit den Zugangsdaten wieder entschlüsselt werden. Dies können Sie selber durchführen oder die Klasse CredentialsDecryptor benutzen, welche sie über die URL

<https://shopproxy.p-s-s.de/CredentialsDecryptor.php.zip> herunterladen können.

**Vor der Einrichtung Ihres Shops müssen Sie uns den AES Key auf sicherem Wege mitteilen.**

## 1.7. Cookies

Vor geraumer Zeit hat das Internet-Konsortium den optionalen Cookie-Parameter [SameSite](#) eingeführt, um zu reglementieren, ob ein Cookie über Domaingrenzen hinweg transferiert werden darf.

Dieser Parameter kann folgende Werte enthalten:

none - Keine Domain-Einschränkungen. (Gleichbedeutend als wäre der Parameter gar nicht gesetzt.)

lax - Cookie-Gültigkeit: Gleiche Top-Level Domain, sowie Subdomains & Unterverzeichnisse

strict - Cookie-Gültigkeit: Gleiche Top-Level Domain

Die Interpretation dieses Parameters wurde nach und nach in die gängigen Web-Browser implementiert, seit kurzem nun auch in den von if-to-shop verwendeten Chromium-Browser.

Da der Shop-Proxy unter einer anderen Top-Level-Domain als der Webshop läuft, werden Cookies mit [SameSite=lax](#) oder [SameSite=strict](#) Browser-seitig nicht bis zum Kundenrechner transferiert, bzw. dort gespeichert.

Für eine reibungslose Kommunikation stellen Sie bitte sicher, dass die von Ihrem Shop verwendeten Cookies diesen Parameter entweder gar nicht, oder nur mit der Einstellung [SameSite=none](#) verwenden.

Um zu Ermitteln ob der Chromium-Browser von if-to-shop verwendet wird, kann der User-Agent des Chromium-Browsers ausgelesen und auf 'if-to-shop' abgefragt werden.

### 1.8. Übertragung der Kommission in einer ZUGFeRD-Rechnung

Wurde beim Übertragen eines Artikels in den Warenkorb eine Kommissionsnummer bzw. Kommissionstext mit übergeben, so sollte diese auf einer Rechnung mit angegeben werden (siehe auch Kapitel 1.5).

Bei der Übertragung der Rechnung im ZUGFeRD 2.0-Format und höher muss die Kommission auf Positionsebene der Rechnung im Block *'IncludedSupplyChainTradeLineItem'* unter *'AdditionalReferenced-Document'* im Tag *'IssuerAssignedID'* eingetragen werden. Zur Kennzeichnung, dass es sich um die Kommission handelt, muss im Tag *'ReferenceTypeCode'* der Code *'AAJ'* für Lieferauftragsnummer stehen.

The screenshot displays the XML structure of a ZUGFeRD invoice. The tree view is expanded to show the 'AdditionalReferencedDocument' element within the 'IncludedSupplyChainTradeLineItem' block. The 'IssuerAssignedID' element is highlighted in blue.

```

EN 16931 – FacturX 1.0.05 – ZUGFeRD 2.1.1 - COMFORT > Elements > CrossIndustryI
S EN 16931 – FacturX 1.0.05 – ZUGFeRD 2.1.1 - COMFORT
> EN 16931 – FacturX 1.0.05 – ZUGFeRD 2.1.1; base
> XSD Data types
> XML Attributes
> CII_QDT_100pD16B; ZF2.1.1
> CII Reusable ABIE RDM Guide; ZF2.1.1
> CII_UDT_100pD16B; ZF2.1.1
▼ Elements *
  ▼ E CrossIndustryInvoice *
    ▼ s xs:sequence *
      > E ExchangedDocumentContext *
      > E ExchangedDocument *
      ▼ E SupplyChainTradeTransaction *
        ▼ s xs:sequence *
          ▼ E IncludedSupplyChainTradeLineItem **
            ▼ s xs:sequence *
              > E AssociatedDocumentLineDocument *
              > E SpecifiedTradeProduct **
              > E SpecifiedLineTradeAgreement **
              > E SpecifiedLineTradeDelivery *
              ▼ E SpecifiedLineTradeSettlement *
                ▼ s xs:sequence *
                  > E ApplicableTradeTax **
                  > E BillingSpecifiedPeriod *
                  > E SpecifiedTradeAllowanceCharge
                  > E SpecifiedTradeSettlementLineMonetarySummation **
                  ▼ E AdditionalReferencedDocument **
                    ▼ s xs:sequence
                      E IssuerAssignedID
                      E TypeCode
                      E ReferenceTypeCode
  
```

## 2. Anhang

### 2.0. Änderungshistorie

<p><b>Änderungen zu Version 1.0.0:</b></p> <ul style="list-style-type: none"> <li>• Der Parameter <a href="#">articleGroup</a> wurde als optionaler Parameter zur Erstaufnahme von Artikeln und zur Bearbeitung von Artikeln hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.0.1:</b></p> <ul style="list-style-type: none"> <li>• Der Bereich <a href="#">descriptiveParameters</a> wurde hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.0.2:</b></p> <ul style="list-style-type: none"> <li>• Der Versionscode wird dem Shop bei jedem Aufruf übergeben</li> </ul>
<p><b>Änderungen zu Version 1.0.3:</b></p> <ul style="list-style-type: none"> <li>• Der Parameter <a href="#">customerNumber</a> wurde hinzugefügt</li> <li>• Der Parameter <a href="#">serverSiededParameters</a> wurde hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.13:</b></p> <ul style="list-style-type: none"> <li>• AES Verschlüsselung und Parmeter <a href="#">aesKey</a> wurden hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.14:</b></p> <ul style="list-style-type: none"> <li>• Der Parameter <a href="#">gtinNumber</a> wurde hinzugefügt</li> <li>• Der Parameter <a href="#">parameterHash</a> wurde hinzugefügt</li> <li>• Der Parameter <a href="#">commissionNumber</a> wurde bei der Übergabe und der Rückgabe zum Warenkorb hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.15:</b></p> <ul style="list-style-type: none"> <li>• Der Parameter <a href="#">parameterHash</a> wurde der Preisabfrage hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.16:</b></p> <ul style="list-style-type: none"> <li>• Die Suche nach Artikeln auf Basis der GTIN Nummer wurde hinzugefügt.</li> </ul>
<p><b>Änderungen zu Version 1.17:</b></p> <ul style="list-style-type: none"> <li>• Der Parameter <a href="#">deliveryDate</a> wurde bei der Übergabe und der Rückgabe zum Warenkorb hinzugefügt</li> <li>• Kapitel 'Vorauswahl eines Artikels mittels GTIN' entfernt</li> </ul>
<p><b>Änderungen zu Version 1.18:</b></p> <ul style="list-style-type: none"> <li>• Fehlerbehandlung implementiert</li> <li>• Kapitel 'Cookies' um User-Agent-Beschreibung erweitert</li> </ul>
<p><b>Änderungen zu Version 1.19:</b></p> <ul style="list-style-type: none"> <li>• Kapitel 'Übertragung der Kommission in einer ZUGFeRD-Rechnung' hinzugefügt</li> </ul>
<p><b>Änderungen zu Version 1.21:</b></p> <ul style="list-style-type: none"> <li>• Kapitel 1 um Absatz mit .htaccess-Einstellungen ergänzt.</li> <li>• Kapitel 1.1, 1.4 und 1.5 um Absatz mit Verpackungseinheit ergänzt.</li> <li>• Kapitel 1.1: <a href="#">longDesc</a> und <a href="#">graphic</a> als notwendig gekennzeichnet.</li> <li>• Kapitel 1.5: Anzahl in Menge geändert.</li> <li>• Kapitel 1.6: Absatz mit dataTest-Verzeichnis hinzugefügt.</li> </ul>
<p><b>Änderungen zu Version 1.22:</b></p> <ul style="list-style-type: none"> <li>• Kapitel 1.4 um Absatz mit dem Content-Type ergänzt.</li> </ul>
<p><b>Änderungen zu Version 1.23:</b></p> <ul style="list-style-type: none"> <li>• Kapitel 1.6: Angaben zur 64 Bit-Version entfernt.</li> </ul>