

Beispiel URL:

<https://username:passwort@domain.com/api/version?token=facec4b71d2434b2f6b7c7e4c5571c4c529973ff&type=xml>

Die API liefert ihre Daten in UTF-8 Kodierung und nach Möglichkeit mit einer "gzipped" Kompression, um die Bandbreite optimal auszunutzen. Um auszuschließen, dass Daten doppelt übertragen werden, sollten sog. eTags verwendet werden.

Für jeden API-Aufruf wird ein Token mitgeschickt, um den Benutzer eindeutig zu identifizieren. Ein Token besteht idealerweise aus einem mit SHA-1 gehashten Passwort, z.B.: facec4b71d2434b2f6b7c7e4c5571c4c529973ff.

Die konkrete Authentifizierung läuft jedoch über HTTP. Zur Übertragung wird HTTPS vorgeschlagen.

Die API liefert sowohl JSON als auch XML. Das zu verwendende Format wird mit Hilfe des type Parameters angegeben. Wird dieser Parameter weggelassen, ist standardmäßig XML zu verwenden.

Als Datumsformat wird ein Unixtimestamp vorgeschlagen. Neben den ansonsten angegebenen HTTP-Statuscodes kann jeder API-Aufruf natürlich einen 500er oder 403er Code liefern.

Stammdaten:

=====

Kunden:

.....

```
<?xml version="1.0">
<customers>
  <customer id="Integer">
    <name>String</name>
    <firstname>String</firstname>
    <addresses>
      <address id="Integer" type="Integer">
        <city>String</city>
        <street>String</street>
        <postcode>String</postcode>
        <country>Integer</country>
      </address>
      ...
    </addresses>
    <email>String</email>
    <birthdate>Datetime</birthdate>
  </customer>
  ...
</customers>
```

```
{
  "customers": [
    "customer": {
      "id": "Integer",
      "name": "String",
      "firstname": "String",
      "addresses": [
        "address": {
```

```

        "id": "Integer",
        "type": "Integer",
        "city": "String",
        "street": "String",
        "postcode": "String",
        "country": "Integer"
    }
],
    "email": "String",
    "birthdate": "Datetime"
}
]
}

```

Artikel:

.....

```
<?xml version="1.0">
```

```

<articles>
  <article id="Integer">
    <title>String</title>
    <description>String</description>
    <ean>Integer</ean>
    <attributes type="Integer">
      <attribute>String</attribute>
    </attributes>
    <price>Float</price>
    <currency>String</currency>
  </article>
</articles>

```

```

{
  "articles": [
    "article": {
      "id": "Integer",
      "title": "String",
      "description": "String",
      "ean": "Integer",
      "attributes": [
        "type": "Integer",
        "attribute": "String"
      ],
      "price": "Float",
      "currency": "String"
    }
  ]
}

```

Order:

.....

```
<?xml version="1.0">
```

```

<order id="Integer" customer="Integer" billaddress="Integer" shippingaddress="Integer" status="Integer">
  <positions>
    <position article="Integer" amount="Float" discount="Float" />
  </positions>
</order>

```

```

    ...
  </positions>
</order>

```

Kundendaten lesen:

URL: GET - {apiurl}/customers?token=...&id=all&type=[xml|json](&since=datetime) [201, 204]
 URL: GET - {apiurl}/customers?token=...&id=Integer&type=[xml|json] [201, 404]

```

<?xml version="1.0">
<customers>
  <customer id="1234">
    <name>Mustermann</name>
    <firstname>Max</firstname>
    <addresses>
      <address id="91203123" type="1">
        <city>Berlin</city>
        <street>Berliner Str. 2</street>
        <postcode>15537</postcode>
        <country>Germany</country>
      </address>
    </addresses>
    <email>max@mustermann.com</email>
    <birthdate>1379585955</birthdate>
  </customer>
</customers>

```

```

{
  "customers": [
    "customer": {
      "id": 1234,
      "name": "Mustermann",
      "firstname": "Max",
      "addresses": [
        "address": {
          "id": 91203123,
          "type": 1,
          "city": "Berlin",
          "street": "Berliner Str. 2",
          "postcode": "15537",
          "country": "Germany"
        }
      ],
      "email": "max@mustermann.com",
      "birthdate": 1379585955
    }
  ]
}

```

Artikel lesen:

URL: GET - {apiurl}/articles?token=...&id=all&type=[xml|json](&since=datetime) [201, 204]
 URL: GET - {apiurl}/articles?token=...&id=Integer&type=[xml|json] [201, 404]

```

<?xml version="1.0">

```

```

<articles>
  <article id="1234">
    <title>Hemd</title>
    <description>Ein Hemd.</description>
    <ean>2312312445</ean>
    <attributes type="Size">
      <attribute>L</attribute>
      <attribute>M</attribute>
    </attributes>
    <price>2.12</price>
    <currency>$</currency>
  </article>
</articles>

```

```

{
  "articles": [
    "article": {
      "id": 1234,
      "title": "Hemd",
      "description": "Ein Hemd.",
      "ean": 2312312445,
      "attributes": {
        "type": "Size",
        "attribute": [
          "L",
          "M"
        ]
      },
      "price": 2.12,
      "currency": "$"
    }
  ]
}

```

Bestellungen lesen:

URL: GET - {apiurl}/orders?token=...&id=all&type=[xml|json](&since=datetime) [201, 204]

URL: GET - {apiurl}/orders?token=...&id=Integer&type=[xml|json] [201, 404]

```

<order id="20938243" customer="1234" billaddress="91203123" shippingaddress="91203123" status=
="1">
  <positions>
    <position article="1234" amount="10.0" discount="0.0" />
  </positions>
</order>

```

```

{
  "order": {
    "id": 20938243,
    "customer": 1234,
    "billaddress": 91203123,
    "shippingaddress": 91203123,
    "status": 1,
    "positions": [
      "position": {
        "article": 1234,
        "amount": 10.0,

```

```
        "discount": 0.0
      }
    ]
  }
}
```

Abfragen der API-Version:

URL: GET - {apiurl}/version?token=...&type=[xml|json] [201]