

Planlogi EDI description

General information

Orders transfer via XML with POST.

All fields must be presented even if they are empty.

XML POST output has to be presented like this. Row's block can contain more than one <Row> blocks.

If header (<Header>) is presented without rows it will be ignored. Every order has at least 1 row.

POST must be presented to a given address, it will be sent with AuthCode to the EDI user.

Server will respond in JSON (if necessary we can change it to xml).

PS! It is recommended to test forwarding orders with our test environment first. Below you will find API keys for testing environment and also for LIVE database.

Use application/xml as Content-Type to send the data.

XML can be posted to <http://planlogi.ee/api/orders/postXML> address.

OrderID and PostedByClientCode: **PLANDEMO**

API key is AuthCode:

- DEV (test): c70eab3a-344e-47cd-af95-b9327f18fad4
- LIVE:

```
<Order>
  <AuthCode></AuthCode>
  <NewOrder>1</NewOrder>
  <Header>
    <OrdererID></OrdererID>
    <PostedByClientCode></PostedByClientCode>
    <PostedByUserID></PostedByUserID>
    <Reference1></Reference1>
    <Reference2></Reference2>
    <AlternativeBarcode></AlternativeBarcode>
    <Picking>
      <Name></Name>
      <Street></Street>
      <City></City>
      <ZipCode></ZipCode>
      <Country></Country>
      <Contact>
        <Person></Person>
        <Phone></Phone>
      </Contact>
      <TimeFrom></TimeFrom>
      <TimeTo></TimeTo>
      <Date></Date>
      <Info></Info>
    </Picking>
    <Delivery>
      <Name></Name>
      <Street></Street>
      <City></City>
      <ZipCode></ZipCode>
      <Country></Country>
      <Contact>
        <Person></Person>
        <Phone></Phone>
      </Contact>
      <TimeFrom></TimeFrom>
      <TimeTo></TimeTo>
    </Delivery>
  </Header>
</Order>
```

```
        <Date></Date>
    </Delivery>
    <Kilometers></Kilometers>
    <Stops></Stops>
    <Hours></Hours>
    <Liters></Liters>
    <Dangerous></Dangerous>
    <Clo></Clo>
    <Fragile></Fragile>
    <TemperatureSensitive></TemperatureSensitive>
    <TemperatureFrom></TemperatureFrom>
    <TemperatureTo></TemperatureTo>
    <WaybillComments></WaybillComments>
    <CarrierCode></CarrierCode>
</Header>
<Rows>
    <Row>
        <LineNo></LineNo>
        <Type></Type>
        <Quantity></Quantity>
        <Description></Description>
        <Weight></Weight>
        <Volume></Volume>
        <LoadingMeters></LoadingMeters>
        <Length></Length>
        <Width></Width>
        <Height></Height>
        <ItemPackageId></ItemPackageId>
    </Row>
</Rows>
</Order>
```

Header info (Order > Header)

Even if some fields are empty, they should be included in the order. If number field is filled then commonly the minimum length of the field is 1 character. There is no limitation if not stated otherwise.

XML field name	Data type	Description	Required field (must be filled)	Min. length of the field	Max. length of the field	Places after comma	Can be empty
AuthCode	text	Clients API code	yes				
NewOrder	integer	Value must always be 1	yes	1	1		
Header > OrdererID	text	Client ID code (is found in Planlogi clients databases clients registry)	yes	2			
Header > PostedByClientCode	text	Order forwarders client ID (is derived from Planlogi clients databases clients registry)	yes	2			
Header > PostedByUserID	integer	Order inserter (user ID, derivated from Planlogi clients databases users registry). If this field is left empty then Planlogi will assign automatically user EDI (this is the user connected to different automatic actions in Planlogi)					x
Header > Reference1	text	Reference 1					x
Header > Reference2	text	Reference 2					x
Header > AlternativeBarcode	text	Alternative Barcode					x
Picking > Name	text	Picking > Company	No				x
Picking > Street	text	Picking > Street	No				x
Picking > City	text	Picking > City	No				x
Picking > ZipCode	integer	Picking > Zip Code	Yes				
Picking > Country	text	Picking > Country We use country code (which can be found in Planlogi clients databases countries registry)	Yes				
Picking > Contact > Person	text	Picking > Contact > Person					x

Picking > Contact > Phone	text	Picking > Contact > Phone					x
Picking > TimeFrom	text	Picking > Time From (start)					x
Picking > TimeTo	text	Picking > Picking To(end)					x
Picking > Date	date	Picking > Date (date must be presented in YYYY-MM-DD format)	yes	10	10		
Picking > Info	text	Picking > Info					x
Delivery > Name	text	Delivery > Company	No				x
Delivery > Street	text	Delivery > Street	No				x
Delivery > City	text	Delivery > City	No				x
Delivery > ZipCode	integer	Delivery > Zip Code	Yes	1			
Delivery > Country	text	Delivery > Country We use country code (it can be found in Planlogi countries registry)	Yes	1			
Delivery > Contact > Person	text	Delivery > Contact > Person					x
Delivery > Contact > Phone	text	Delivery > Contact > Phone					x
Delivery > TimeFrom	text	Delivery > Time From (start)					x
Delivery > TimeTo	text	Delivery > Time To (end)					x
Delivery > Date	date	Delivery > Date (date must be presented in YYYY-MM-DD format)		10	10		x
Kilometres	text	Kilometres		1			x
Stops	text	Stops		1			x
Hours	text	Hours		1			x
Litres	text	Litres		1			x
Dangerous	integer	Special Conditions > Dangerous 0 – no, 1 – yes (if empty/null, system will assign value 0)		1	1	0	x
Clo	integer	Special Conditions > Clo 0 – no, 1 – yes (if empty/null, system will assign value 0)		1	1	0	x

Fragile	integer	Special Conditions > Fragile 0 – no, 1 – yes (if empty/null, system will assign value 0)		1	1	0	x
TemperatureSensitive	integer	Special Conditions > Temperature Sensitive 0 – no, 1 – yes (if empty/null, system will assign value 0)		1	1	0	x
TemperatureFrom	text	Special Conditions > Temperature From (start)					x
TemperatureTo	text	Special Conditions > Temperature To (end)					x
WaybillComments	text	Waybill Comments					x
CarrierCode	text	Carrier Code (derivated from Planlogi clients databases client's registry)					x

Orders row info (Order > Rows > Row)

Even if some fields are empty, they should be included in the order.

XML field name	Data type	Description	Required field (must be filled)	Min. length of the field	Max. length of the field	Places after comma	Can be empty
LineNo	integer	Row number (always starts from 1)	Yes	1			
Type	text	Type (can be found in Planlogi clients databases types registry)	Yes				
Quantity	integer	Quantity	Yes	1			
Description	text	Order row description	Yes	1			
Weight	decimal	Weight (kg)	Yes	1		3	
Volume	text	Volume (m3)				3	x
LoadingMeters	text	LDM (m)				2	x
Length	text	Depth/Lenght (cm)				3	x
Width	text	Width (cm)				3	x
Height	text	Height (cm)				3	x
ItemPackageld	text	In case you need to also forward sending systems ID or reference (if you wish to have two way data exchange in the future)					x

Server responses

Successful query will also return inserted order identification code. In return HTTP status codes are 200, 201. Server will return in JSON format whereas information will also include orderAuthCode (inserted orders identification code). Result will be presented in this format:

```
{"type": "success", "result": {"message": "Order accepted", "data": {"orderAuthCode": "f2aa8d76ee98857e916489b5b678988e"}}
```

Failed query will return HTTP status code 400, 404 or 500. Result will be presented in JSON format.

For example if error occurs on xml schema validation

```
{"type": "error", "result": {"message": "Failed to validate XML", "detail": "[\" element Header: Schemas validity error : Element 'Header': Character content other than whitespace is not allowed because the content type is 'element-only'.\"]"}}
```

Error code 400 with the message “Bad request” is presented when there is no insert or xml is missing element Order and/or AuthCode. This error code can also be presented in case there is no API key or the API key in POST is not active anymore.

```
{"type": "error", "result": {"message": "API key issue", "detail": "Provided API key doesn't exist or is not active"}}
```

Error code 404 with the message “Failed to validate” XML is presented if the XML does not validate with given schema (<http://planlogi.ee/schemas/ordersPostXML.xsd>)

Error code 500 with the message “Failed to create temporary file” is presented if in Planlogi server experiences failure to save the XML content.

Examples

Python:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import requests

xml = "<?xml version='1.0' encoding='UTF-8'?><Order><AuthCode>c70eab3a-344e-47cd-af95-
b9327f18fad4</AuthCode><NewOrder>1</NewOrder><Header><OrdererID>PLANDEMO</OrdererID><PostedByClientCode>PLANDEMO</PostedByClientC
ode><PostedByUserID></PostedByUserID><Reference1>viide1 info</Reference1><Reference2>viide2
info</Reference2><AlternativeBarcode></AlternativeBarcode><Picking><Name>Planlogi</Name><Street>Jakobsoni
11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person>Pirjo</Person><Phone>123 456
789</Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-
07</Date><Info></Info></Picking><Delivery><Name>Planlogi</Name><Street>Jakobsoni
11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person></Person><Phone></Phone></Contact><
TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-
07</Date></Delivery><Kilometers></Kilometers><Stops></Stops><Hours></Hours><Liters></Liters><Dangerous></Dangerous><Clo></Clo><Fr
agile></Fragile><TemperatureSensitive></TemperatureSensitive><TemperatureFrom></TemperatureFrom><TemperatureTo></TemperatureTo><W
aybillComments>see on
test</WaybillComments><CarrierCode></CarrierCode></Header><Rows><Row><LineNo>1</LineNo><Type>EUR</Type><Quantity>1</Quantity><Des
cription>...</Description><Weight>2</Weight><Volume></Volume><LoadingMeters>0.4</LoadingMeters><Length></Length><Width></Width><H
eight></Height><ItemPackageId></ItemPackageId></Row><Row><LineNo>2</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...
</Description><Weight>3</Weight><Volume></Volume><LoadingMeters>0.42</LoadingMeters><Length></Length><Width></Width><Height></Hei
ght><ItemPackageId></ItemPackageId></Row></Rows></Order>"

# set headers which is requested by server
headers = {'Content-Type': 'application/xml'}

postlink = 'http://planlogi.ee/api/orders/postXML'
print requests.post(postlink, data=xml, headers=headers).text
```

Curl (command line):

```
curl -v -X POST -H 'Content-Type: application/xml' -d '<?xml version="1.0" encoding="UTF-8"?><Order><AuthCode>c70eab3a-344e-47cd-af95-b9327f18fad4</AuthCode><NewOrder>1</NewOrder><Header><OrdererID>PLANDEMO</OrdererID><PostedByClientCode>PLANDEMO</PostedByClientCode><PostedByUserID></PostedByUserID><Reference1>viide1 info</Reference1><Reference2>viide2 info</Reference2><AlternativeBarcode></AlternativeBarcode><Picking><Name>Planlogi</Name><Street>Jakobsoni 11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person>Pirjo</Person><Phone>123 456 789</Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-07</Date><Info></Info></Picking><Delivery><Name>Planlogi</Name><Street>Jakobsoni 11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person></Person><Phone></Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-07</Date></Delivery><Kilometers></Kilometers><Stops></Stops><Hours></Hours><Liters></Liters><Dangerous></Dangerous><Clo></Clo><Fragile></Fragile><TemperatureSensitive></TemperatureSensitive><TemperatureFrom></TemperatureFrom><TemperatureTo></TemperatureTo><WaybillComments>see on test</WaybillComments><CarrierCode></CarrierCode></Header><Rows><Row><LineNo>1</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...</Description><Weight>2</Weight><Volume></Volume><LoadingMeters>0.4</LoadingMeters><Length></Length><Width></Width><Height></Height><ItemPackageId></ItemPackageId></Row><Row><LineNo>2</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...</Description><Weight>3</Weight><Volume></Volume><LoadingMeters>0.42</LoadingMeters><Length></Length><Width></Width><Height></Height><ItemPackageId></ItemPackageId></Row></Rows></Order>' http://planlogi.ee/api/orders/postXML
```

Node.js:

```
const http = require('http');

const postData = '<?xml version="1.0" encoding="UTF-8"?><Order><AuthCode>c70eab3a-344e-47cd-af95-b9327f18fad4</AuthCode><NewOrder>1</NewOrder><Header><OrdererID>PLANDEMO</OrdererID><PostedByClientCode>PLANDEMO</PostedByClientCode><PostedByUserID></PostedByUserID><Reference1>viide1 info</Reference1><Reference2>viide2 info</Reference2><AlternativeBarcode></AlternativeBarcode><Picking><Name>Planlogi</Name><Street>Jakobsoni 11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person>Pirjo</Person><Phone>123 456 789</Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-07</Date><Info></Info></Picking><Delivery><Name>Planlogi</Name><Street>Jakobsoni 11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person></Person><Phone></Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-07</Date></Delivery><Kilometers></Kilometers><Stops></Stops><Hours></Hours><Liters></Liters><Dangerous></Dangerous><Clo></Clo><Fragile></Fragile><TemperatureSensitive></TemperatureSensitive><TemperatureFrom></TemperatureFrom><TemperatureTo></TemperatureTo><WaybillComments>see on test</WaybillComments><CarrierCode></CarrierCode></Header><Rows><Row><LineNo>1</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...</Description><Weight>2</Weight><Volume></Volume><LoadingMeters>0.4</LoadingMeters><Length></Length><Width></Width><Height></Height><ItemPackageId></ItemPackageId></Row><Row><LineNo>2</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...</Description><Weight>3</Weight><Volume></Volume><LoadingMeters>0.42</LoadingMeters><Length></Length><Width></Width><Height></Height></ItemPackageId></ItemPackageId></Row></Rows></Order>';

let http_opts = {
  hostname: 'planlogi.ee',
  path: '/api/orders/postXML',
  method: 'POST',
  headers: {
    'Content-Type': 'application/xml',
    'Content-Length': Buffer.byteLength(postData)
  }
};

const req = http.request(http_opts, res => {
  console.log('status: ', res.statusCode);
  console.log('headers: ', JSON.stringify(res.headers));

  res.setEncoding('utf8');
  res.on('data', chunk => {
    console.log('body: ', chunk);
  });

  res.on('end', () => {
    console.log('request finished');
  });
});
```

```
});  
  
req.on('error', e => {  
  // e.message  
  console.log('error: ', e);  
});  
  
req.write(postData);  
req.end();
```

PHP:

```
<?php
    $url = 'http://planlogi.ee/api/orders/postXML';

    $xml = '<?xml version="1.0" encoding="UTF-8"?><Order><AuthCode>c70eab3a-344e-47cd-af95-
b9327f18fad4</AuthCode><NewOrder>1</NewOrder><Header><OrdererID>PLANDEMO</OrdererID><PostedByClientCode>PLANDEMO</PostedByClientCode><PostedByUserID></PostedByUserID><Reference1>viide1 info</Reference1><Reference2>viide2
info</Reference2><AlternativeBarcode></AlternativeBarcode><Picking><Name>Planlogi</Name><Street>Jakobsoni
11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person>Pirjo</Person><Phone>123 456
789</Phone></Contact><TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-
07</Date><Info></Info></Picking><Delivery><Name>Planlogi</Name><Street>Jakobsoni
11</Street><City>Viljandi</City><ZipCode>71020</ZipCode><Country>EE</Country><Contact><Person></Person><Phone></Phone></Contact><
TimeFrom></TimeFrom><TimeTo></TimeTo><Date>2018-06-
07</Date></Delivery><Kilometers></Kilometers><Stops></Stops><Hours></Hours><Liters></Liters><Dangerous></Dangerous><Clo></Clo><Fr
agile></Fragile><TemperatureSensitive></TemperatureSensitive><TemperatureFrom></TemperatureFrom><TemperatureTo></TemperatureTo><W
aybillComments>see on
test</WaybillComments><CarrierCode></CarrierCode></Header><Rows><Row><LineNo>1</LineNo><Type>EUR</Type><Quantity>1</Quantity><Des
cription>...</Description><Weight>2</Weight><Volume></Volume><LoadingMeters>0.4</LoadingMeters><Length></Length><Width></Width><H
eight></Height><ItemPackageId></ItemPackageId></Row><Row><LineNo>2</LineNo><Type>EUR</Type><Quantity>1</Quantity><Description>...
</Description><Weight>3</Weight><Volume></Volume><LoadingMeters>0.42</LoadingMeters><Length></Length><Width></Width><Height></Hei
ght><ItemPackageId></ItemPackageId></Row></Rows></Order>';

    $context = stream_context_create(array(
        'http' => array(
            'method' => 'POST',
            'header' => 'Content-type: application/xml',
            'content' => $xml,
            'timeout' => 60
        )
    ));

    $resp = file_get_contents($url, false, $context);

    var_dump($resp);

?>
```