Pubhub Web Services

Third Party Sales Data API Integration Guide

April 2017
v1.0

TABLE OF CONTENTS

REVISION HISTORY

Date	Version	Comment
2017-04-19	1.0	First edition

OVERVIEW

This API is intended for third party receivers who receives metadata from Pubhub and who is able to deliver sales data back to Pubhub to ease publishers processing of sales data. Publishers usually receives sales data from many different retailers and in many different formats. By delivering sales data in a uniform way to Pubhub publishers only have to receive sales data from one source (Pubhub).

Invoicing between third party and a publisher can either be handled by Publizon or happen directly between third party and publisher. This is configured by the publisher in the Pubhub backend.

HOW TO ACCESS THE WEB SERVICE

The Sales Data API exists in two separate environments, each with their own server, code and database. Please use the web service located at **sales.qa.pubhub.dk** when developing and testing! Please request a new license key for the production environment when you're ready to go live.

Test / development environment:
http://sales.qa.pubhub.dk/v1/

Production environment:
https://sales.pubhub.dk/v1/

Attention: Use SSL/https in production!

ENDPOINT DESCRIPTIONS

/TRANSACTIONS

Post sales data (in the form of atomic transactions).

Test: POST http://sales.qa.pubhub.dk/api/v1/transactions
Prod: POST https://sales.pubhub.dk/api/v1/transactions

Request headers

Parameter name	Type	Description
x-license-key	String	Client's license key (please request this from Publizon)

Request body

A list of atomic transactions each representing a single sale / order (quantity > 0 and <=1) or cancellation (quantity >= -1 and < 0).

Parameter name	Type	Description
TransactionId	String	Unique id to identify the transaction. Max 50 characters.
CancelTransactionId	String	Id to identify an existing transaction to be cancelled. Max 50 characters. Leave empty for non-cancellation transactions.
CreatedUtc	Date	Creation time of the order. Must be in UTC in the following format YYYY-MM-DDTHH:MM:SSZ (e.g. 2017-04-21T14:00:00Z).
FulfilledUtc	Date	Fulfilled/payment time of the order. Must be in UTC in the following format YYYY-MM-DDTHH:MM:SSZ (e.g. 2017-04-21T14:00:00Z).
Isbn13	String	ISBN-13 of the book.
Price	Decimal	Amount paid to the publisher after any discounts.
Quantity	Decimal	Quantity/reading percentage of the order. Use a value of 1 if your system does not operate with quantities. A quantity value of 0.2 would depict that 20% of the book has been read.
CurrencyCode	String	ISO 4217 currency code. Only DKK is supported at the moment.

Example

```
[
 {
  "transactionId": "12345",
  "createdUtc": "2017-04-21T14:00:00Z",
  "fulfilledUtc": "2017-04-21T14:00:00Z",
  "isbn13": "9788740006469",
  "price": 100,
  "quantity": 1,
  "currencyCode": "DKK"
 },
{
  "transactionId": "12346",
  "cancelTransactionId": "54321",
  "createdUtc": "2017-04-21T14:00:00Z",
  "fulfilledUtc": "2017-04-21T14:00:00Z",
  "isbn13": "9788740006469",
  "price": -10,
  "quantity": -1,
  "currencyCode": "DKK"
 }
]
```

Response

The response contains the following feedback for each individual transaction in the request.

```
{
 "status": "[Status]",
 "transactionId": "[TransactionId]",
 "errors": [Errors],
 "warnings": [Warnings]
}
```

where

[Status] = Success, Error or Warning
[TransactionId] = The transaction id from the request
[Errors] = null (if no errors) or a list (1..n) of errors (see error codes below)
[Warnings] = null (if no warnings) or a list (1..n) of warnings (see warning codes below)

Example

```
[
 {
    "status": "Success",
    "transactionId": "123",
    "errors": null,
    "warnings": null
 },
 {
    "status": "Warning",
    "transactionId": "1234",
    "errors": null,
    "warnings": {"transactionId": ["PUB-003: 'Transaction Id' already exists."]}
 },
 {
    "status": "Error",
    "transactionId": "12345",
    "errors":
    {
      "currencyCode": ["PUB-032: 'Currency Code' must be DKK."],
      "quantity": ["PUB-025: 'Quantity' must be greater than '0'. For cancellations, 'CancelTransactionId' must be specified."],
      "price": ["PUB-073: 'Price' must be greater than '0'. For cancellations, 'CancelTransactionId' must be specified."]
    },
    "warnings": null
 }
]
```

Error and warning codes

Code	Type	Description
PUB-001	Error	'Transaction Id' should not be empty.
PUB-002	Error	'Transaction Id' must be between 1 and 50 characters. You entered X characters.
PUB-003	Warning	'Transaction Id' already exists.
PUB-011	Error	'Isbn13' should not be empty.
PUB-012	Error	'Isbn13' is not a valid ISBN-13.
PUB-013	Error	'Isbn13' is invalid; content does not exist.
PUB-021	Error	'Quantity' should not be empty.
PUB-022	Error	'Quantity' must be less than '0' for cancellation request.
PUB-023	Error	'Quantity' must be greater than or equal to '-1' for cancellation request.
PUB-024	Error	'Quantity' is invalid; value does not match the value of the transaction that is being cancelled.
PUB-025	Error	'Quantity' must be greater than '0'. For cancellations, 'CancelTransactionId' must be specified.
PUB-026	Error	'Quantity' must be less than or equal to '1'.
PUB-031	Error	'CurencyCode' should not be empty.
PUB-032	Error	'Currency Code' must be DKK.
PUB-041	Error	'Created Utc' should not be empty.
PUB-051	Error	'Fulfilled Utc' is invalid; sales data for the previous calendar month can only be submitted until the 1. Of the month 12:00 PM.
PUB-061	Error	'Cancel Transaction Id' is invalid; transaction does not exist.
PUB-071	Error	'Price' must be less than '0' for cancellation request.
PUB-072	Error	'Price' is invalid; value does not match the value of the transaction that is being cancelled.
PUB-073	Error	'Price' must be greater than '0'. For cancellations, 'CancelTransactionId' must be specified.