

## Publizon Offline API

### REVISION

2014-03-02 - v 1.0: Oprettet.

2014-03-05 – v 1.0.1: Domæner i url'er tilføjet

### BEGREBER / FORMATTER

- **(E-BØGER) DE-KEY (Data Encryption Key):** Denne nøgle er unik for platformen samt applikations versionen. Nøglen bruges af API såvel som applikationen til at kryptere/dekryptere bog pakker. Krypteringsalgoritmen der benyttes er Rijndael (AES) algoritmen. Disse AES indstillinger benyttes:
  - **Blocksize: 128.**
  - **Feedbacksize: 128**
  - **Nøglestørrelse: 256**
  - **Mode: CBC**
  - **Padding PKCS7**Nøgle og vector (IV) udledes af Publizon. Ved oprettelse af ny nøgle kræves følgende data udleveret:
  - **Platform (f.eks. Apple iOS)**
  - **Applikationsversion (f.eks. 1.0.1)**
  - **Virksomhedsnavn.***Ved download lydpakker krypteres data ikke og der skal derfor ikke bruges denne nøgle til dekryptering af data.*
- **(EBØGER) SP (Stream Package):** Stream pakke. Denne pakke indeholder bog data.
- **(EBØGER) Platform:** Ved API requests hvor platform navn er en nødvendighed angives platformen der matcher enheden ud fra denne liste:
  - **Apple iPhone/iPad: ios**
  - **Microsoft Windows Phone: winphone**
  - **Microsoft Windows Tablet: wintablet**
  - **Microsoft Windows PC: winpc**
  - **Apple MacOSX: macosx**
  - **Google Android: android**
- **Version:** Alle versions numre angives i format: [MAJOR].[MINOR].[BUILD] - f.eks.: 1.2.0
- **(LYDBØGER) Sekvens:** En sekvens er en del af en lydbog bestående af ca. en times afspilnings data.
- **ContentVersionOrder:** Dette ID er Publizons interne reference til slutbrugerens lån/køb.
- **SERVICE KEY:** Ved API requests skal service key altid tilføjes i headeren. Service nøglen er den nøgle som blev udleveret da virksomheden blev oprettet som kunde hos Publizon.

### **(E-BØGER) DOWNLOAD AF E-BØGER TIL OFFLINE LÆSNING:**

For at danne en stream pakke til offline læsning sendes følgende request:

**REQUEST FRA ENHED:**

HEADER:

X-SERVICE-KEY: (Service Key)

HTTPS GET

offline.api.streaming.pubhub.dk/package/v1/{content\_version\_order\_id}?app\_version={app\_version}&platform={platform}

BODY: (EMPTY)

Der vil nu blive dannet en stream pakke hvis indhold vil blive krypteret med den krypteringsnøgle/encoding der matcher "app\_version" samt "platform" værdierne:

**RESPONSE FRA API**

HEADER:

CONTENT-TYPE: text/xml

BODY:

<package>

<downloadsource>{uri\_til\_stream\_pakke}</downloadsource>

</package>

**(LYDBØGER) DOWNLOAD AF LYDBØGER TIL OFFLINE LÆSNING:**

En lydbog består af et bestemt antal sekvenser af ca. en times varighed. For at download en eller flere sekvenser gøres følgende:

Først skal klienten have et overblik over hvor mange sekvenser den konkrete order består af - dette gøres ved at sende følgende request:

**REQUEST FRA ENHED:**

HEADER:

X-SERVICE-KEY: (Service Key)

HTTP GET offline.api.streaming.pubhub.dk/v2/playlist/{content\_version\_order\_id}

BODY: (EMPTY)

**RESPONSE FRA API**

HEADER:

CONTENT-TYPE: application/json

BODY:

{

"duration":{total\_playlist\_duration},

"expiredate": {expire\_date\_utc},

"tracks":[

```
{
  "sequence":{sequence_index},
  "position":{sequence_start_index_in_seconds},
  "duration":{sequence_duration_in_seconds}
}
]
```

**Duration:** Den total længde af lydbogen (angivet i sekunder).

**ExpireDate:** Udløbsdato for lånet (ikke udfyldt med ordre uden udløbsdato)

**Tracks:** Array af sekvenser.

**Sequence:** Sekvens indeks.

**Position:** Start positionen for sekvensen (Angivet i sekunder)

**Duration:** Længde på sekvensen (Angivet i sekunder).

Når enheden har kendskab til antal sekvenser en order består af kan en enkelt sekvens downloades ved at lave følgende request:

**REQUEST FRA ENHED:**

**HEADER:**

**X-SERVICE-KEY:** (Service Key)

**HTTPS GET** *offline.api.pubhub.dk/v2/package/{content\_version\_order\_id}/{sequence\_index}*

**BODY:** (EMPTY)

**RESPONSE FRA API**

**HEADER:**

**CONTENT-TYPE:** *audio/mpeg*

**BODY:**

*(MPEG3 DATA)*

**(E-BØGER) EN STREAM PAKKES "ANATOMI":**

En stream pakke fil (Filendelse: \*.spkg) er en pakkefil som består af bog data.

Filen kan åbnes ved at benytte sig af ZIP dekomprimering.

I pakken vil du finde filer som f.eks. billeder, HTML filer samt XML filer dog har de alle den ting til fælles at de har fået en ekstra filendelse tilknyttet: \*.spkgi.

Alle spkgi filer er krypteret og skal dekrypteres før data kan indlæses.

En dekrypteret fil må kun leve i enhedens hukommelse (RAM) og ikke på dens lagringsenhed (harddisk)

I pakken finder du filen også "PURCHASE.XML.SPkgI" denne XML fil indeholder data omkring købet/lånet – XML attributten "expire" er datoen for hvornår lånet ophøre – skal kun bruges ved lån.

```
<purchase cvo="{content_version_order_id}" date="{purchase_date_time_utc}" expire="{expire_date_time_utc}"
/>
```

### **(LYDBØGER) EN LYDBOGS "PAKKE":**

En lydbooks "pakke" hentes ved at danne følgende request:

#### **REQUEST FRA ENHED:**

HEADER:

X-SERVICE-KEY: (Service Key)

HTTPS GET audio.api.streaming.pubhub.dk/v1/orders/{content\_version\_order\_id}

BODY: (EMPTY)

#### **RESPONSE FRA API**

HEADER:

CONTENT-TYPE: application/json

BODY:

```
{
  "artist": {author},
  "availableoffline":true,
  "cover": {uri_to_book_cover},
  "duration":{total_duration_in_seconds},
  "state":{
    "nightmode":{endusers_nightmode_state},
    "position":{endusers_current_cursor_position}
  },
  "title":{book_title}
}
```

**Artist:** Navnet på forfatteren bag bogen

**Availableoffline:** Boolsk værdi der angiver om bogen kan downloades til offline brug

**Cover:** URI til bogens forside.

**Duration:** Bogens afspilningstid i sekunder.

**Nightmode:** Boolsk værdi der fortæller om slutbrugeren har aktiveret "nat læsning"

**Position:** Værdi der hvor langt slutbrugeren er kommet i lydbogen (sekunder henne i bogen).

**Title:** Bogens titel

### **(E-BØGER) OPDATERING AF JAVASCRIPT READER:**

Skulle der opstå et behov for at opdatere den javascript baseret reader kan dette kun gøres når enheden er online. Opdateringer kan downloades ved at danne følgende request:

#### **REQUEST FRA ENHED:**

HEADER:

X-SERVICE-KEY: (Service Key)

HTTPS GET

offline.api.streaming.pubhub.dk/version/v1/check/readerjs/?app\_version={app\_version}&platform={platform}&js\_version={javascript\_version}

BODY: (EMPTY)

#### RESPONSE FRA API

HEADER:

CONTENT-TYPE: text/xml

BODY:

<version>

<latest>{true|false}</latest>

<src>{url\_to\_update\_download}</src>

</version>

Skulle det ikke være seneste udgave af javascript readeren der er installeret på enheden vil "latest" være "false" og "src" vil indholde et URI til den nyeste opdatering – URI'et vil have et format som dette:

<http://download.streaming.pubhub.dk/jsreader/reader-1.0.2.min.js>

#### (EBØGER) MÆRKER (NOTER, BOGMÆRKER OG HUSKE MÆRKER):

Der findes tre typer mærker som læser applikationen skal arbejde med:

**Note: Et mærke notat.**

**Bookmark: Et mærke der angiver hvor slutbrugeren er kommet til.**

**Readingstate: Et mærke der husker hvor slutbrugeren er kommet til i bogen.**

For at hente de mærker som måtte være tilknyttet en ordre kaldes APIet:

#### REQUEST FRA ENHED:

HEADER:

X-SERVICE-KEY: (Service Key)

CONTENT-TYPE: application/json eller text/xml

HTTP GET

offline.api.streaming.pubhub.dk/v2/marks/?order={content\_version\_order\_id}

BODY: (EMPTY)

#### RESPONSE FRA API

HEADER:

CONTENT-TYPE: application/json (eller text/xml hvis forespurgt)

BODY:

```
[
  {
    "cvo":{content_version_order_id},
    "created":{creation_date_utc},
    "id": {unique_marking_id},
    "type":{type_of_mark},
```

```
"tocindex":{table_of_content_index},
"startcharindex":{start_character_index},
"endcharindex":{end_char_index},
"message":{message_in_mark}
}
]
```

**Cvo\*:** Det unikke order id.

**Created\*:** Oprettelses datoen for mærket (UTC).

**Id\*:** Det unikke id for mærket.

**Type\*:** Mærke type ("bookmark", "note" eller "readingstate").

**TocIndex\*:** "Table of content" indeks.

**StartCharIndex\*:** Karakter start indeks på den angivet side.

**EndCharIndex:** Karakter afslutning på den angivet side (Kun ved typen noter)

**Message:** Besked hæftet til mærket.

**\*= Ikke null værdier.**

#### REQUEST FRA ENHED:

HEADER:

X-SERVICE-KEY: (Service Key)

CONTENT-LENGTH: Body længde

CONTENT-TYPE: application/json eller text/xml

HTTP POST

offline.api.streaming.pubhub.dk/v2/marks/

BODY:

```
<?xml version="1.0" encoding="utf-8"?>
<marks xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<mark>
<cvo>e3919b79-c249-41f7-b70a-37790beb41ab</cvo>
<created>2013-07-09T11:55:46.153</created>
<id>2921093f-603e-433f-b4f9-fb16d1468b59</id>
<type>bookmark</type>
<persistencestate>create</persistencestate>
<tocindex>5</tocindex>
<startcharindex>1</startcharindex>
<message>hello world</message>
</mark>
</marks>
```

**Cvo\*:** Det unikke order id.

**Created\*:** Oprettelses datoen for mærket (UTC).

**Id\*:** Det unikke id for mærket.

**Type\*:** Mærke type ("bookmark", "note" eller "readingstate").

**TocIndex\*:** "Table of content" indeks.

**StartCharIndex\*:** Karakter start indeks på den angivet side.

**EndCharIndex:** Karakter afslutning på den angivet side (Kun ved typen noter)

**Message: Besked hæftet til mærket.**

**PersistenceState\*:** Mærkets status på klient enheden (“deleted”, “update” eller “create”)

**\*= Ikke null værdier.**

API’et vil nu synkronisere klientens mærker med de allerede lageret mærker og returnere resultatet:

#### RESPONSE FRA API

HEADER:

CONTENT-TYPE: application/json (eller text/xml hvis forespurgt)

BODY:

```
[
  {
    "cvo":{content_version_order_id},
    "created":{creation_date_utc},
    "id": {unique_marking_id},
    "type":{type_of_mark},
    "tocindex":{table_of_content_index},
    "startcharindex":{start_character_index},
    "endcharindex":{end_char_index},
    "message":{message_in_mark}
  }
]
```

Ved synkronisering vil API’et flage det seneste oprettet mærke som vinder af synkroniseringen (Set ud fra værdierne “created” og “persistencestate”) medmindre en invalid forespørgesel bliver lavet - f.eks.: opdatering af mærke som ikke allerede findes i database - skulle dette ske vil mærket blive ignoreret af API’et.

Hvis der allerede findes et readingstate mærke i databasen på order “XYZ” og klienten sende et nyt readingstate med persistencestate “create” for samme order vil API’et kaste en fejl.

Der må kun være ET readingstate mærker pr. order.

#### (LYDBØGER) MÆRKER (POSITION, NATTELÆSNING):

Ved lydbøger kan der via API’et hentes informationer om hvor langt slutbrugeren har afspillet sin bog samt slutbrugers indstillinger for afspilleren - dette gøres ved at sende et request til API’et:

#### REQUEST FRA ENHED:

HEADER:

X-SERVICE-KEY: (Service Key)

CONTENT-TYPE: application/json

HTTP GET

audio.api.streaming.pubhub.dk/v1/states/{content\_version\_order\_id}

BODY: (EMPTY)

#### RESPONSE FRA API

HEADER:

CONTENT-TYPE: application/json

BODY:

```
{"nightmode":true,"position":30}
```

**Nightmode:** Boolsk værdi der siger om slutbrugerens afspiller er i natte læsning eller ej.

**Position:** Værdi der fortæller hvor slutbrugerens afspiller er kommet til ved sidste afspilning (Angivet i sekunder)

For at gemme slutbrugerens afspilnings position samt afspiller indstillinger sendes dette request til API'et:

#### REQUEST FRA ENHED:

HEADER:

X-SERVICE-KEY: (Service Key)

CONTENT-TYPE: application/json

HTTP POST

audio.api.streaming.pubhub.dk/v1/states/{content\_version\_order\_id}

BODY:

```
{"nightmode":true,"position":30}
```

#### OFTE STILLEDE SPØRGSMÅL

- **Hvad gør jeg hvis jeg har mistanke om at en DE-KEY eller SERVICE-KEY er blevet kompromitteret:**  
Skulle der være mistanke om at en DE-KEY og/eller en SERVICE-KEY er blevet misbrugt kontaktes Steffen Sørensen via. e-mail: [scs@publizon.dk](mailto:scs@publizon.dk).