



Bookrepublic ExLibris

Documentazione tecnica

Versione 0.7

Introduzione

Per cominciare

Bookrepublic permette agli store online di collegarsi al proprio repository, chiamato ExLibris, attraverso un sistema di lanci settimanale e attraverso delle API dedicate:

- Lanci: tutte le informazioni relative al catalogo dei titoli disponibili - e le loro rispettive variazioni - verrà comunicato attraverso il meccanismo di lanci via FTP nel formato standard *ONIX for books 3.0*.¹ Il catalogo Bookrepublic sarà fornito interamente alla prima attivazione. Nei successivi lanci saranno esportati solo i record relativi alle variazioni di catalogo (nuovi ebook presenti, ebook di cui è variato qualche dato, ebook cancellati dal repository di Bookrepublic);
- API: attraverso il sistema di API sarà possibile generare gli ebook a transazione avvenuta e consultare lo storico delle transazioni eseguite. Questa funzione non permette altre interrogazioni.

Cosa serve?

All'accreditamento dello store presso la piattaforma ExLibris, Bookrepublic fornirà alcuni dati necessari alle librerie per dialogare con il nostro repository:

- SECRET e KEY necessari ad autenticare le richieste API (secondo quanto spiegato successivamente);
- l'accesso (con USERNAME e PASSWORD) alla piattaforma <https://exlibris.bookrepublic.it/>. Attraverso questo canale sarà possibile per le librerie visualizzare gli editori attivi, generare un nuovo SECRET e una nuova KEY, modificare la propria password e gestire il proprio account in maniera autonoma;
- per ricevere i dati del catalogo ebook di Bookrepublic attraverso il sistema di lanci, sarà invece necessario comunicare un indirizzo FTP o sFTP (comprensivo di USERNAME e PASSWORD) presso il quale ricevere l'esportazione dei dati di catalogo in formato ONIX 3.0.

¹ <http://www.editeur.org/12/About-Release-3.0/>

API

Introduzione

ExLibris espone una API basata su HTTP che permette agli utilizzatori di interagire con la piattaforma. Nella versione 1.0 le librerie possono recuperare lo storico delle proprie transazioni e acquistare nuovi ebook.

L'API di ExLibris utilizza HTTPS come protocollo applicativo per garantire la riservatezza delle trasmissioni e un meccanismo di firma basato su HMAC e SHA1 per garantire l'autenticazione dell'operatore.

Key, secret, marker e signature

L'accesso alle API di ExLibris viene autorizzato solo se ogni richiesta HTTP viene firmata digitalmente dal client; il meccanismo di autorizzazione si basa su un dato pubblico (la key) e su uno privato (il secret). Quest'ultimo dato è condiviso tra la libreria ed ExLibris e non deve mai essere divulgato.

Ogni richiesta HTTP deve contenere nella url la key e nella query string almeno tre valori: timestamp, marker e sig:

Esempio di url:

```
https://exlibris.bookrepublic.it/api/ffffffff-ffff-ffff-ffff-fffffffffffff/?timestamp=1284752328&marker=somestuff&sig=xxx
```

timestamp è il tempo, espresso come unix timestamp UTC arrotondato al secondo, in cui la libreria effettua la richiesta. Per evitare i reply attack, il webserver di ExLibris rifiuta richieste con timestamp futuro o al di fuori di una certa finestra temporale, per questo motivo è importante che il client **sia sincronizzato** con un server NTP affidabile. La finestra temporale al momento è impostata a 30 secondi, questo valore potrà cambiare in futuro.

marker è un valore alfanumerico completamente sotto il controllo del client che identifica **in maniera univoca** la singola richiesta. ExLibris tiene traccia dei marker utilizzati da ogni api key e impedisce ogni successivo riutilizzo dello stesso valore.

sig è la signature della richiesta, client e server la calcolano in maniera autonoma utilizzando il segreto condiviso, se i due valori coincidono la richiesta viene autorizzata.

Calcolo delle signature

La signature viene calcolata come segue:

1. Viene costruito un messaggio M utilizzando i parametri della query string (a eccezione di sig) e del body che vengono concatenati con i rispettivi valori; le stringhe così ottenute vengono a loro volta concatenate tra loro secondo l'ordine alfabetico.

Il messaggio M costruito sulla url di esempio precedente è

`markersomestufftimestamp1284752328`

Regole per la costruzione del messaggio M:

- il parametro timestamp viene incluso in M;
- la key, facendo parte della url, non viene inclusa in M;
- se un parametro ha più valori, questi vengono concatenati al primo secondo l'ordine in cui sono trasmessi;

Esempio:

`https://.../?param=value1¶m=value3¶m=value2`

M = paramvalue1value3value2

- parametri e valori vengono inseriti in M prima di essere stati codificati per poter fare parte di una URL (url encoding);
- i parametri devono essere codificati in ascii mentre i valori in utf-8.

2. Viene generata una signature S utilizzando HMAC con il secret condiviso e la funzione di hashing SHA1; la rappresentazione esadecimale di S viene usata come valore per il parametro sig.

Nota bene: al momento l'unico Content-type supportato per le richieste è application/x-www-form-urlencoded.

Start URL e risposte

L'API di ExLibris è costruita in modo tale che per un client sia necessario conoscere solo una url di partenza; da questa, analizzando la risposta, può ricostruire le url per accedere alle funzionalità offerte dal server.

Le risposte generate dal server di ExLibris sono sempre documenti JSON.

La starting url è `https://exlibris.bookrepublic.it/api/{KEY}/` dove al posto di {KEY} bisogna sostituire la propria api key; la risposta a questa richiesta è un object (secondo la terminologia JSON) che ha per attributi i nomi delle funzionalità esposte dalla API e per valori le url a cui richiederle.

Un esempio di risposta può essere:

```
{  
  
'transactions':  
'https://exlibris.bookrepublic.it/api/ffffffff-ffff-ffff-ffff-fffffffffff  
ff/transactions/'  
  
}
```

questo oggetto elenca una funzionalità, transactions, e la url per utilizzarla.

Retrocompatibilità

Il web service è pensato per evolversi nel tempo rimanendo compatibile con client scritti in precedenza.

In futuro ExLibris potrà offrire funzionalità aggiuntive, che andranno a popolare l'oggetto ritornato dalla starting url, restando compatibile con i client creati in precedenza.

Le funzioni offerte dal web service sono da considerarsi stabili, viene garantito che la semantica dell'operazione, l'elenco e il significato dei parametri non varierà nel tempo. Una funzionalità potrà evolversi in futuro supportando nuovi parametri, a patto che nessuno di questi sia obbligatorio né che l'utilizzo di questi nuovi parametri modifichi lo scopo originale della funzionalità.

Il valore di ritorno di una funzionalità potrà essere esteso per supportare nuovi attributi ma non potrà mai essere ridotto.

Reference

Transactions

Utilizzando transactions è possibile accedere allo storico con le transazioni effettuate o acquistare un nuovo ebook. Sono supportati i metodi GET e POST.

L'accesso allo storico avviene utilizzando il metodo GET, non sono previsti parametri obbligatori e la risposta consiste in un array di object; gli object hanno il seguente layout:

```
{  
  
  'id': '*id transazione exlibris*',  
  
  'ebook': {  
  
    'isbn': '*isbn ebook*',  
  
    'title': '*titolo ebook*',  
  
    'url': '*url per il download dell'ebook*'   
  
  },  
  
  'buyer_transaction_id': '*id associato dalla libreria*',  
  
  'customer_name': '*nome del cliente finale (se fornito durante l'acquisto)*',  
  
  'customer_email': '*email del cliente finale (se fornito durante l'acquisto)*',  
  
  'message': '*messaggio del cliente finale (se fornito durante l'acquisto)*',  
  
  'price': '*prezzo dell'ebook*',
```

```
'date': '*data di creazione della transazione (formato ISO)*'  
}
```

L'acquisto di un nuovo ebook avviene utilizzando il metodo POST, la richiesta deve contenere (nel body) i parametri:

- isbn: l'isbn del libro da acquistare;
- buyer_transaction_id: un identificativo utilizzato dalla libreria per identificare questa transazione (max 25 caratteri);
- customer_name: il nome del cliente finale;
- customer_email: l'email del cliente finale che verrà inclusa nell'exlibris dell'ebook;
- message: un messaggio, tipo dedica, da includere con la transazione;

La risposta consiste in un singolo object con i dati della transazione (stesso formato degli oggetti ritornati dal GET).

Gestione delle transazioni

Transazioni

Dopo aver importato i dati forniti da Bookrepublic relativamente al catalogo, la libreria - a transazione avvenuta - potrà generare il rispettivo ebook interrogando il nostro repository attraverso una API debitamente compilata.

Lo storico delle transazioni avvenute e degli ebook generati sarà conservato da Bookrepublic in un repository dedicato per ogni libreria. Ogni singola libreria potrà decidere in autonomia, attraverso la propria console, all'indirizzo <https://exlibris.bookrepublic.it/>:

- per quanto tempo i file saranno disponibili (ad esempio: 6 mesi);
- quante volte il file potrà essere scaricato (ad esempio: fino a un limite di 15 download).

Per trovare le informazioni

Bookrepublic ha aperto un repository pubblico all'indirizzo http://bitbucket.org/bookrepublic/client_exlibris/src. All'interno di questo spazio è possibile:

- accedere alla documentazione di supporto costantemente aggiornata;
- trovare client in linguaggio PHP e Python (all'indirizzo http://bitbucket.org/bookrepublic/client_exlibris/src/tip/client/);
- segnalare eventuali problemi o suggerimenti all'interno dello spazio ISSUE.²

² Creare una nuova ISSUE è semplice: dopo aver cliccato la relativa tab da menù, basterà selezionare *create new* indicando poi un titolo e il relativo contenuto. Una singola ISSUE può essere liberamente commentata, cosicché diventi uno spazio di confronto tra il team di Bookrepublic e le librerie associate.

Revisioni

Versione	Data	Modifiche
ver. 0.6	20 settembre 2010	Aggiunta tabella TAG Onix.
ver. 0.7	21 settembre 2010	Modifiche alle funzionalità API. Modifiche al protocollo.