Inventory Update API
User Documentation

AbeBooks.com®

INTRODUCTION

The Inventory Update API is a service that may be used to update a bookseller's inventory of new, used, rare, and out of print books. The service is designed to be integrated into the bookseller's own custom inventory management system, in order to allow them to make real time updates to their online AbeBooks inventory.

XML API

The following sections describe the minimum requirements, usage and functionality, valid requests, responses, and error codes.

Minimum Requirements

- An AbeBooks bookseller account with a valid username and API Key.
- In-depth understanding of XML. For more information, see http://www.w3.org/XML/
- Appropriate system access to open port 443 for bi-directional TLS traffic.
- Ability to work with and deploy a programming language that supports TLS and XML.
- Because each XML request contains your client PIN and password, you must use a TLS certified secure client.
- Operation through a proxy server is not supported.
- The bookseller must manage unique Book IDs (vendorBookID) from their inventory tool/system.
- When sending an inventory update request, the bookseller must include all book information, as the server will update all fields with each request. The exception to this is delete requests.

Service Address

URL: https://inventoryupdate.abebooks.com
Port: 10027

Security

Security for the service is based on TLS and valid client (vetted bookseller) sign on credentials.

Testing:

AbeBooks has no separate test environment for the Inventory Update server. You are responsible for ensuring that your client software is performing as required. To test your system, you can simply add, delete and update books via the server using your existing bookseller account credentials.

Overview of Service Usage and Functionality

A bookseller will use the inventory update server and its XML API to manage their online book inventory with AbeBooks. The advantage to using the engine is that the bookseller can update their inventory in near real time, resulting in fewer books being sold on AbeBooks that were previously sold by the bookseller, improved fulfillment ratings for the bookseller and better customer service.

There is no software for the bookseller to install, only this specification document that the bookseller will use to create an interface with AbeBooks inventory management server using an appropriate software development language that supports TLS and XML.

The API has only one method, the inventoryUpdateRequest. The request includes action and AbebookList elements. The AbebookList may contain up to 100 books (Abebook). Each Abebook element includes a transactionType of "add," "update" or "delete."

The bookseller creates XML documents that conform to the API specification, references the included XML schema document to validate the request and submits the request to the server. An appropriate success or error response will be returned.

There is a requirement for the bookseller to manage unique Book IDs in their system and send that ID with the request in the vendorBookID tag.

If an "add" or "update" request is sent for a book with a vendorBookID that exists in the bookseller's online inventory, the book will be updated with the new information. If any tags are empty or omitted, any existing information in the corresponding fields will be removed.

Use Cases

Bookseller adds a new book

To add a new book to the bookseller's online inventory, they send an inventory update request with the transaction type "add".

Book is no longer available for sale on AbeBooks

If the bookseller sells a book from their store that is listed in their AbeBooks online inventory, the bookseller will send an inventory update request with the transaction type "delete". If this same book becomes available at a later date, the bookseller can send an inventory update request with the transaction type "update", and the book will be reinstated. Please note we will delete any fields that are left blank or overwrite them with the new information when an "add" or "update" request is sent for an existing vendorBookID

Bookseller updates a book

If the bookseller needs to update a book listed in their online inventory, they send an inventory update request where the transaction type is "update". They must include all the book information.

Bookseller wishes to reinstate a book that was previously deleted

A bookseller can reinstate a book that has been deleted or sold on AbeBooks by sending an inventory update request where the transaction type is "update".

XML Requests

This section describes an XML request made to the Inventory Update server. The first section covers common request elements. The next sections describe the XML request in detail.

Request Elements

All inventory update requests follow XML standards and should include the standard XML header:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

The following text contains an example of an inventoryUpdateRequest with a TransactionType of "add", and a Quantity of "2".

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<inventoryUpdateRequest version="1.0">
 <action name="bookupdate">
  <username>BOOKSELLER_1</username>
  <password>20DigitAPPKey</password>
 </action>
 <AbebookList>
  <Abebook>
   <transactionType>add</transactionType>
   <vendorBookID>000001</vendorBookID>
   <author>Wittgenstein, Ludwig</author>
   <title>Tractatus Logico-Philosophicus</title>
   <publisher>Dover Publications</publisher>
   <subject>Philosophy Logic Linguistics</subject>
   <price currency="USD">1234567.89</price>
   <dustJacket>true</dustJacket>
   <binding type="hard">Leather</binding>
   <firstEdition>false</firstEdition>
   <signed>false</signed>
   <booksellerCatalogue>Philosophy</booksellerCatalogue>
   <description>Clean and unmarked; with original intro.</description>
   <bookCondition>Fine</bookCondition>
   <size>8.4" x 5.4"</size>
   <jacketCondition>Fine</jacketCondition>
   <bookType>Ex\-Library</bookType>
   <isbn>0486404455</isbn>
   <publishPlace>Toronto, Canada, ON</publishPlace>
   <publishYear>1999</publishYear>
   <edition>Reprint</edition>
   <inscriptionType>Signed by previous owner.</inscriptionType>
   <quantity amount="2"></quantity>
   <pictureList>
    <pictureURL>https://www.abebooks.com/images/123456_1.jpg</pictureURL>
    <pictureURL>https://www.abebooks.com/images/123456_2.jpg</pictureURL>
   </pictureList>
  </Abebook>
 </AbebookList>
</inventoryUpdateRequest>
```

XML Code 4-1 XML Elements common to all requests

inventoryUpdateRequest Tag

This tag encompasses the entire inventory update request. It has one attribute that indicates the version of the XML request and two elements: AbebookList and Abebook. Currently the only valid version is "1.0".

action Tag

This tag tells the inventory server which action to take and which user is executing the action. The tag has one attribute and two elements: The attribute "name" may have the value of "bookupdate" indicating which action to take. The value of the elements username and password must identify a unique, valid bookseller account, where username is the bookseller's client PIN.

AbebookList Tag

The AbebookList element must contain one Abebook element and can contain up to a maximum of 100 Abebook elements.

Abebook Tag

The Abebook tag contains all of the elements that describe a book. transactionType, vendorBookID and price are required. One of author, title or publisher is also required. The remaining elements are optional as described in the tag dictionary.

Quantity tag

Specifying the quantity of a book can be done with the "quantity" tag. This tag has two attributes: "amount", and "limit". The following example illustrates how to specify a quantity of "2" using the "amount" attribute:

```
<quantity amount="2"></quantity>
```

If you have an unlimited quantity of certain books, you can set them to "unlimited" by using the "limit" attribute. The "limit" attribute must be either a "1", or a "0". Here is an example of how to specify an unlimited quantity of a book:

```
<quantity limit="1"></quantity>
```

The Inventory Update Response

The inventoryUpdateRequest will return a response code and message in the inventoryUpdateResponse for each add, update or delete. Errors will be returned as described in Figure 4.3.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <inventoryUpdateResponse version="1.0">
  <code>600</code>
  <message lang="en">English Message</message>
  <AbebookList>
  <Abebook>
   <code>600</code>
   <message>Book added or updated<message>
   <transactionType>ADD<transactionType>
   <vendorBookID>8484848E8e8e8<vendorBookID>
  </Abebook>
  <Abebook>
   <code>600</code>
   <message>Book deleted</message>
   <transactionType>DELETE</transactionType>
   <vendorBookID>1212121</vendorBookID>
  </Abebook>
  <Abebook>
   <code>600</code>
   <message>Book added or updated</message>
   <transactionType>UPDATE</transactionType>
   <vendorBookID>1212121</vendorBookID>
  </Abebook>
   .
   .
   .
  </AbebookList>
</inventoryUpdateResponse>
```

XML Code 4-3 Example InventoryUpdateResponse

XML Error Responses

The server can respond with a number of error messages. These messages are divided into API responses and generic server error responses, which are listed in the Appendix A. Error responses take the following format:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<requestError version="1.0">
    <code>501</code>
    <message lang="en">English Error Message</message>
</requestError>
```

XML Code 4-4 Format of an Error Response

Response Codes

The following table outlines the possible response codes that may be returned by the Inventory Update server, along with the action that should be taken if the error is received.

Code	Message	Solution
600	Successful transaction	No action required
601	Book id is not valid	Modify the criteria you specified in your request. Ensure that books have been included in the request and the vendorBookID field is no more than 15 characters long and contains valid data.
602	Book list too large	Decrease the number of books in the request so that it is less than or equal to the maximum.
603	Illegal Book Transaction	Ensure that all books have the transactionType field set to ADD, UPDATE or DELETE.
604	Bad Price	Ensure that the value in the price field is a valid positive number.
606	Bad Book Data	Ensure that all required fields are filled and that at least one of the author, title and publisher fields contains useful data.

Table 4-5 Response Error Codes

APPENDIX

Appendix A: General Server Error Codes

AbeBooks.com uses a generic XML server framework that may return the error codes below.

Code	Message	Solution
100	Database Unavailable	A database error has occurred at AbeBooks.com. Contact AbeBooks.
103	Network Error	Some form of network error has occurred. Try again later.
104	Invalid XML	The XML sent to the server is invalid
105	Unhandled exception	Try again later
106	External Server Unavailable	Try again later
108	Unauthorized Action	You are attempting to perform an action you do not have permission to perform
109	Unknown Action	You have sent an action that is not valid. Correct your XML request and retry.
110	User is invalid. Either it is unknown or has an incorrect password	Correct the user name and password in your XML request
111	There was an error validating the user	The server was unable to validate the user in the action. Try again later.

Appendix B: Tag Dictionary

The following table outlines all of the tags that are available to be used in an Inventory Update request.

Tag	Homebase	Required for Add/Upd/Del	Field Size	Notes
<InventoryUpdateRequest>	N/A	Yes/Yes/Yes	N/A	1.0 is the only valid version.
<action name>	N/A	Yes/Yes/Yes	N/A	The action name is a required attribute and identifies the request as a book update request. "bookupdate" is the only valid value.
<username/>	N/A	Yes/Yes/Yes	30	
<password/>	N/A	Yes/Yes/Yes	30	
</action>	N/A	Yes/Yes/Yes	N/A	
<AbebookList>	N/A	Yes/Yes/Yes	N/A	
<Abebook>	N/A	Yes/Yes/Yes	N/A	
<transactionType/>	N/A	Yes/Yes/Yes	N/A	ADD, UPDATE or DELETE are the only valid values. (Not case sensitive)
<vendorBookID/>	BOOK	Yes/Yes/Yes	32	Bookseller's vendorBookID must be Unique.
<author/>	ANAM	Yes/Yes/No	750	On add or update transactions author or title or publisher is required
<title/>	TNAM	Yes/Yes/No	750	
<publisher/>	PBLS	Yes/Yes/No	750	
<subject/>	SUB1	No/No/No	2000	KEYWORDS field
<price>	PRIC	Yes/Yes/No	8.2	The Currency attribute is the three-letter ISO code (for example, "USD"). The price must be a number and have the following format: #.##
<dustJacket/>	JACK	No/No/No	Bool	TRUE/FALSE
<binding>	BIND and BNDC	No/No/No	30	Valid values for Type are "hard" or "soft".
<firstEdition/>	EDTN	No/No/No	Bool	TRUE/FALSE

<signed/>	SIGN	No/No/No	Bool	TRUE/FALSE
<booksellerCatalogue/>	BCAT	No/No/No	750	If the bookseller catalogue supplied in an add or update is "Sold" the update will delete the book.
<description/>	DESC	No/No/No	4000	
<illustrator/>	INAM	No/No/No	254	
<bookCondition/>	BOOC	No/No/No	30	See Keywords Document for values.
<size/>	SIZE	No/No/No	50	
<jacketCondition/>	JCKC	No/No/No	30	See Keywords Document for values.
<bookType/>	BTYP	No/No/No	30	See Keywords Document for values.
<isbn/>	ISBN	No/No/No	15	
<publishPlace/>	PBPL	No/No/No	50	
<publishYear/>	PBYR	No/No/No	4	
<editionList/>	N/A	No/No/No	N/A	Use this tag as parent to specify multiple editions using <edition/> tag
<edition/>	EDNT	No/No/No	40	See Keywords Document for values.
<inscriptionType/>	SGNT	No/No/No	50	See Keywords Document for values.
<quantity/>	N/A	No/No/No	999	999 is the maximum. Limit allows the vendor to specify unlimited quantity by setting limit=1. If amount=0 the listing will be deleted.
<pictureList>	N/A	No/No/No	N/A	Parent tag for a sequence of <pictureURL> items. List may contain up to 5 URLs. Any additional URLs are ignored.
<pictureURL>	N/A	No/No/No	2000	URLs from Amazon, eBay, or other sites where you are unlikely to hold the image copyright are not permitted and will be blocked.
<productionMethod/>	N/A	No/No/No	N/A	This tag supports a 'P' string value, indicating a book is printed on demand, and an 'E' string value, indicating an eBook. Other values will be ignored.
<productType/>	N/A	No/No/No	N/A	See keywords doc for values.
</Abebook>	N/A	Yes/Yes/Yes	N/A	
</AbebookList>	N/A	Yes/Yes/Yes	N/A	
</inventoryUpdateRequest>	N/A	Yes/Yes/Yes	N/A	