

VITAL SOURCE TECHNOLOGIES THIRD-PARTY STORE INTEGRATION

DEVELOPER'S GUIDE

Last updated: May 15, 2013

Vital Source Technologies, Inc.

INGRAM.



VitalSource®

1. Introduction

This document provides the technical framework for inserting fulfillment from an external ecommerce store into a customer's LMS (such as Blackboard) and the various VitalSource systems for licensing and viewing content. This document outlines steps that are needed to integrate a 3rd party eCommerce site with the VitalSource Business Center. Those steps are: *get inventory, add to cart, check out, fulfill the purchase, and return the user.*

At a high-level, the VitalSource API (V3) is a REST-inspired XML over HTTP interface where HTTP status codes are used in lieu of XML errors where appropriate. Occasionally, some newer APIs use the JSON format. An API Key must accompany every request and all requests must be sent using the HTTPS protocol. There is complete documentation for these APIs available, but this document is intended to be inclusive of everything one would need to know to integrate.

The primary domains for this API are `api.vitalsource.com` and `bc.vitalsource.com`. For testing your most recent changes, please use the `api-dev.vitalsource.com` and `bc-dev.vitalsource.com`, and use `api-staging.vitalsource.com` and `bc-staging.vitalsource.com` is available as a customer integration endpoint, since it contains a copy of production code, but with a different test database.

2. Getting Inventory

An *Available Inventory Service* (see Appendix) is available to enumerate which products that may be distributed by a designated API key.

Example Request:

Header: X-VitalSource-API-Key: YOURKEY

GET `https://api.vitalsource.com/v3/products/available.xml`

Example Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
```

```
<product name="Strategies for Successful Lawn Bowling" format="VitalBook"
author-name="Johnson, Rick" role="distributor" isbn13="" print-isbn=""
list-price="0.0" eisbn="" type="Single" publisher="DeVry Migration"
publisher-ref="" digital-list-price="0.0" sku="ABCDEFGHJIJ" description=""
build-status="Live" toc="" cover-image-
url="http://covers.vitalbook.com/vbid/ABCDEFGHJIJ/width/120" print-drm-
text="This title will only allow printing of 30 consecutive pages at a
time." copy-drm-text="This title will only allow copying of 2 consecutive
pages at a time." edition="2">
</product>
...
</products>
```

The products that are returned are those which the client company owns or has been granted the right to distribute. The SKU is especially important, as this will be used in fulfilling the purchase in a later step. The other metadata supplied is helpful for the eCommerce site to display the product.

3. Add to Cart

The eCommerce site needs to have a cart service that can receive web POST's from our fulfillment service. Users that are prompted to access the book, but have not yet paid for the book, will be directed to your eCommerce site to purchase the license, and VitalSource will POST this information to your server.

When VitalSource POSTs this information to your service, your service will need to accept and store a parameter called *bc_token*. Immediately upon receiving this request, the 3rd party store should make a POST callback to https://bc.vitalsource.com/store?bc_token=1a2b3c4d531a2b3c4d53 (appending the *bc_token* passed to you earlier).

This web request (callback) will return a JSON response body with these attributes:

- **course:** A course identifier
- **return_url:** A url to which the user will be redirected when a successful purchase has been made
- **institution:** An institution name
- **guid:** A user identifier that will be used to identify a registered user in your system.
- **role:** - *student* or *instructor*
- **apikey:** - a VitalSource API key to use for fulfillment
- **vbid:** - The VitalSource id of the item being purchased. This will match SKU in available inventory.

Example Request:

Header: X-VitalSource-API-Key: YOURKEY

POST https://bc.vitalsource.com/store?bc_token=1a2b3c4d531a2b3c4d53

Example Response:

```
{
  "course": "456434513",
  "return_url": "http://bc.vitalsource.com/books/L-999-70200?bc_token=5659e0f0-5484-0130-4924-002436f3cae6",
  "institution": "3PSTest",
  "guid": "AAFAAFDFDFDAFDDAAADFF",
  "role": "student",
  "apikey": "VVBBAACDDAAFFAA",
  "vbid": "L-999-70200"
}
```

After completing this call and stashing the information retrieved, it is expected that your store would display the selected item in a shopping cart page. The *bc_token* expires within a couple of minutes after issue so it should be immediately used upon receipt.

4. Checking Out

The user should be taken through the normal checkout process. If the user decides to cancel the purchase at any time, please redirect them to the *return_url* (which was specified in the JSON callback) with *completed=false* appended to the URL.

5. Fulfilling the Purchase

Once the user has successfully paid for their purchase, the 3rd party store will need to provision a redemption code for the user for the particular title and redeem that code on the user's behalf.

To create a redemption code, use the Code Create API (see Appendix).

Example Request:

Header: X-VitalSource-API-Key: YOURKEY

POST <https://api.vitalsource.com/v3/codes.xml>

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<codes sku="0471430749" license-type="perpetual" num-codes="1" online-
  license-type="numdays" online-num-days="365" />
```

Example Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<codes>
  <code>XYZABCDEFBUZGHDKKABD</code>
</codes>
```

The *code* attribute in the response will be a valid, unused redemption code in the VitalSource system which will, when redeemed, grant the redeeming user access to the title specified during code creation.

The final step of the fulfillment process is to redeem the code on behalf of the user. For this, use the Redemption API (See Appendix) to associate the code which was provisioned above with the user. The value which is used to identify the user is called the user GUID. This value (*guid*) was retrieved at the beginning of the process in the JSON package.

Example Request:

Header: X-VitalSource-API-Key: YOURKEY

POST <https://api.vitalsource.com/v3/redemptions.xml>

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<redemption>
  <user>guidvaluegoeshere</user>
  <code>FKKZT47FW4C4S7SNXR3Z</code>
</redemption>
```

Example Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <item href="vbk: 0471430749" title="Britannica Concise Encyclopaedia"
    author="Encyclopaedia Britannica"></item>
</library>
```

6. Returning the User

Once the user has completed their purchase (and the products have been fulfilled successfully), the user will be returned to VitalSource / the originating LMS system. At that point, the purchased book will be opened in their web browser (assuming that provisioning was successful). The 3rd party store should redirect the user's browser to the *return_url* (received in the JSON package) with *completed=true* appended.

7. Error Conditions

Each individual API service (inventory, code provisioning, and code redemption) all have specific error codes defined in the Appendix. If the user explicitly cancels the order or an unexpected error returns, they should be offered the opportunity to return to VitalSource at the `return_url` with `completed=false`.

8. Appendix

1. Available Inventory API

Provides a listing of all products that are either owned or distributable by the company of the API client.

- **URL:** <https://api.vitalsource.com/v3/products/available.xml>
- **Formats:** XML
- **HTTP Method:** GET
- **Request Body:** None
- **Requires user authentication:** false
- **Headers:** `X-VitalSource-API-Key: ABCDEFGHIJKLMNOP`
- **Response:**

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
  ...more results here...
  <product name="Strategies for Successful Lawn Bowling"
    format="VitalBook" author-name="Johnson, Rick" role="distributor"
    isbn13="" print-isbn="" list-price="0.0" eisbn="" type="Single"
    publisher="DeVry Migration" publisher-ref="" digital-list-price="0.0"
    sku="ABCDEFGH IJ" description="" build-status="Live" toc="" cover-
    image-url="http://covers.vitalbook.com/vbid/ABCDEFGH IJ/width/120"
    print-drm-text="This title will only allow printing of 30 consecutive
    pages at a time." copy-drm-text="This title will only allow copying
    of 2 consecutive pages at a time." edition="10">
  </product>
```

```
</products>
```

- **Notes:**
The result set is filtered by what the company of the API user owns or is allowed to distribute.

2. Create Code(s) API

Generate one or more redemption codes

- **URL:** <https://api.vitalsource.com/v3/codes.xml>
- **Formats:** XML
- **Requires user authentication:** false
- **HTTP Method:** POST
- **Headers:** X-VitalSource-API-Key: ABCDEFGHIJKLMNOP
- **Request Body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<codes sku="0471430749" license-type="perpetual" num-codes="2" online-
license-type="numdays" online-num-days="365" />
```
- **Response Body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<codes>
  <code>XYZABCDEFBUZGHDKKABD</code>
  <code>BCDBZDEFCDEBUAARGHZE</code>
</codes>
```
- **Fields:**
 1. *sku* (required) - product identifier referenced by the code
 2. *license-type* (required) - perpetual, numdays, absdate, noaccess, default
 3. *num-codes* (required) - the number of codes of this type being requested
 4. *num-days* (required for numdays) - the number of days this license should be valid for, after redemption
 5. *exp-year* (required for absdate) - expiration year for the absolute date
 6. *exp-month* (required for absdate) - expiration month for the absolute date
 7. *exp-day* (required for absdate) - expiration day for the absolute date
 8. *online-license-type* (optional) - perpetual, numdays, absdate, noaccess
 9. *online-num-days* (required for numdays) - the number of days the online license should be valid for, after redemption

10. *online-exp-year* (required for *absdate*) - online expiration year for the absolute date
 11. *online-exp-month* (required for *absdate*) - online expiration month for the absolute date
 12. *online-exp-day* (required for *absdate*) - online expiration day for the absolute date
 13. *tag* (optional) - mark a batch of codes with a particular code for tracking purposes. colons and spaces are not permitted in tags. it is suggested that tags be 10 characters or fewer, but there is a 50 character hard limit.
 14. *code-type* (optional) - *comp*, *add-drop*, *p-plus-e*, *demo*, *temp*, and *sample* are valid values
 15. *price* (optional) - custom price for code, must be > 0.00
- **Notes:**
 1. The *license-type* option refers to the downloadable license for the book.
 2. The *default* type option for downloadable is new in V3. Instead of specifying a specific license rule for the code, it relies on the default rule for the product. Omitting the *online-type* will use the product default online access setting.
 3. The *noaccess* license type option provides a way to restrict an access method (downloadable or online).
 4. If the caller does not have distribution rights to the particular title, they will receive an HTTP 403 (permission denied)
 5. Hard expiration dates cannot be in the past.

3. Redeem Code for a User

Redeem a code for a particular VitalSource user account

- **URL:** <https://api.vitalsource.com/v3/redemptions.xml>
- **Formats:** XML
- **Requires user authentication:** false
- **HTTP Method:** POST
- **Headers:** X-VitalSource-API-Key: ABCDEFGHIJKLMNOP
- **Request Body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<redemption>
  <user>damon@vitalbook.com</user>
  <code>FKKZT47FW4C4S7SNXR3Z</code>
</redemption>
```
- **Response Body:**

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<library>
  <item href="vbk:1-59339-248-6" title="2005 Britannica Concise
    Encyclopaedia" author="Encyclopaedia Britannica"></item>
</library>
```

- **Fields:**

User: either the email address or the GUID on file for the user's account.

Code: A valid, unexpired VitalSource redemption code

Notes:

- **If the access token header is set, that user value will override the user value in the XML.**
- Authentication is not required to add content to a user's account, thus there is no access token requirement in the call.
- The <library> response is named that way for legacy reasons. It contains references to the content items which were just activated as a result of the redemption.

XML RESPONSE CODES:

Code	Message
468	Redemption code is invalid
469	Code has been redeemed
470	Redemption failed
474	User could not be found
483	Malformed redemption request

SUCCESS EXAMPLE: (where user is identified by the XML element):

POST: /v3/redemptions.xml (Header: no access token header passed)

BODY:

```
<?xml version="1.0" encoding="UTF-8"?>
<redemption>
  <user>damon@vitalbook.com</user>
  <code>FKKZT47FW4C4S7SNXR3Z</code>
</redemption>
```

RESPONSE:

HTTP RESPONSE CODE: 200 (Ok)

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
```

```
<item href="vbk:1-59339-248-6" title="2005 Britannica Concise  
Encyclopaedia" author="Encyclopaedia Britannica"></item>  
</library>
```