

# Bookleteer API

Version: 0.6

Date	Version	Author	Comments
23-02-2011	0.2	Yasir Assam	Added error <b>PDF_GENERATION_ERROR</b> Corrected example code for editEBook call
28-02-2011	0.3	Yasir Assam	<b>MISSING_PARAMETER</b> error report now includes the missing parameter in <b>extraError</b> Added note about parameters being case sensitive <b>UNRECOGNISED_PARAMETER</b> error added
01-07-2011	0.4	Yasir Assam	Added <b>bookreader</b> , <b>authorBio</b> , <b>summary</b> , <b>publisherName</b> & <b>copyright</b> parameters to the <b>editEBook</b> call <b>BITLY_LINK_ERROR</b> and <b>INVALID_NUM_PAGES</b> error codes added The <b>listPublications</b> call now returns fields <b>bookreader</b> and <b>bookreaderLink</b> for each publication that has set these. Added a new call <b>getBookreader</b>
12-07-2011	0.4a	Yasir Assam	<b>NO API CHANGE</b> – this is only a documentation update – the API hasn't changed Added clarification for the <b>isPublic</b> parameter.
08-02-2012	0.4b	Yasir Assam	<b>NO API CHANGE</b> – this is only a documentation update – the API hasn't changed (although a bug was fixed) Added description to <b>createImageFromUpload</b> section
10-08-2012	0.5	Yasir Assam	<b>getBookreader</b> – param <b>key</b> is now optional
17-08-2012	0.6	Yasir Assam	<b>getPublicationPDF</b> - param <b>key</b> is now optional

## Introduction

**NOTE:** All parameter names are case-sensitive, e.g. if a function expects **id** and you pass **ID** the call will fail.

Every API call requires the **key** parameter, in which you pass your Bookleteer API key, a 40 character hexadecimal string which you can find on [http://bookleteer.com/my\\_account.html](http://bookleteer.com/my_account.html) (only Pro members are given an API key).

So in the list of API calls below, bear in mind that every parameters section includes the **key** parameter in addition to the parameters listed (for calls that only have the **key** parameter, no parameters section is given).

Because every API call takes the **key** parameter, every API call can return one of the following errors:

- **MISSING\_PARAMETER** – the API key was not provided (this error can be sent back if any required parameter is missing).
- **NOT\_AUTHORIZED** – the API key does not correspond to a known user, or the user with this API key isn't authorised to use the API.

Also, every call can return the **UNRECOGNISED\_PARAMETER** error.

The result of each API call is either a JSON object or some other files type (e.g. a PDF file, a JPEG image). For method calls that don't normally return a JSON object (e.g. **getPublicationPDF**, which returns a PDF), a JSON will be returned on an error. You'll need to detect the mime type or content type to tell whether an error is returned (then check the JSON object for the error message).

JSON objects are returned in a compact form (no newlines). If you want a human-readable version you can put the JSON code through a formatter, e.g. <http://jsonformat.com> or <http://jsonformatter.curiousconcept.com/>.

Every JSON result has the following format:

```
{
  "result": "OK",
  "payload": ...
}
```

The **payload** field (which could be missing) contains the data result of the API call, and the **result** field contains an error message (or “OK” if there was no error). Some errors also include an **extraError** field alongside the **result** and **payload** fields.

Every API call uses the URL format <http://bookleteer.com/api/method>, where *method* is the method name, e.g. **listPublications**.

The API calls come in two types:

- **GET**. e.g. <http://bookleteer.com/api/getPublication?key=123&id=332&pageSize=a4>
- **POST**. Use HTML 'form' parameters

GET calls are used when there is no side-effect when making the call. POST is used when making the call creates a side-effect. For example, **getImage** has no side-effects (it just returns an image), whereas **deleteImage** causes an image to be delete, hence it uses POST.

**NOTE:** You can test GET calls by using a browser and entering the call in the address bar. To test POST calls you can either create a form in HTML or write code in your favourite language (PHP examples using cURL are used in this doc).

## Errors

Here is a complete list of all the errors that can be returned by the API calls:

- **OK** – No error (only returned in JSON objects, e.g. PDF returns won't have this set, obviously)
- **MISSING\_PARAMETER** – A parameter is missing. **extraError** contains the missing parameters.
- **NOT\_AUTHORIZED** – the API key does not correspond to a known user, or the user with this API key isn't authorised to make this call.
- **PUBLICATION\_NOT\_FOUND** – publication referred to by **id** either doesn't exist or the caller (identified by the **key** parameter) isn't authorised to view it.
- **PAGE\_SIZE\_NOT\_FOUND** - the publication identified by **id** doesn't have a PDF in the page size identified by **pageSize**
- **IMAGE\_NOT\_FOUND** – the image identified by **id** not found.
- **IMAGE\_SIZE\_NOT\_SUPPORTED** – you need to specify one of “full”, “preview” or “thumb” for the image size (any other value will return this error).
- **CANNOT\_GET\_EXTERNAL** – you're trying to get an external URL image, and bookleteer.com doesn't host these images. Use the external URL to fetch the image.
- **IMAGE\_FILE\_NOT\_FOUND** – bookleteer.com couldn't find the request image file in its cache. This error shouldn't happen – if it does please report it to bookleteer.com.

- **CURL\_ERROR** – cURL was used to fetch an image and it reported an error. When you get this error, the **extraError** parameter of the returned JSON object contains the specific cURL error (number and string equivalent).
- **HTTP\_ERROR** – the server returned an HTTP code that was an error (e.g. 404). The exact error number is given in the **extraError** parameter.
- **NOT\_IMAGE** – file uploaded or referred to by URL is not an acceptable image format
- **FILE\_UPLOAD\_ERROR** – PHP file upload error. The **extraError** parameter contains an **errNo** field which is one of the codes found here: <http://www.php.net/manual/en/features.file-upload.errors.php>
- **IMAGE\_IN\_USE** – you can't delete this image as it's in use in one of more of your publications
- **BAD\_DATE\_FORMAT** – you need to use the format YYYY-MM-DD, e.g. 2011-02-28
- **BAD\_DESIGN\_ID** – the design you specified either does not exist, or is not suitable for the publication you're trying to create (e.g. you're trying to use an eBook design with a StoryCube).
- **BAD\_ORIENTATION** – orientation value not recognised – you need to pass either “landscape” or “portrait”
- **WRONG\_IMAGE\_TYPE** – you're trying to use a PDF when a bitmap is expected or vice-versa
- **BAD\_WRITING\_DIRECTION** – something other than “ltr” (left-to-right) or “rtl” (right-to-left) was specified
- **BAD\_FORMAT** – something other than “classic” or “book” was specified
- **WRONG\_PUBLICATION\_TYPE** – you're trying to retrieve data for a publication which doesn't use this type, e.g. the publication is a StoryCube and you're trying to get its HTML data.
- **PDF\_GENERATION\_ERROR** – unable to generate PDF. A more detailed error message is given in **extraError**
- **UNRECOGNISED\_PARAMETER** – at least one parameter (given in **extraError**) is not recognised for this call
- **BITLY\_LINK\_ERROR** – unable to generate bit.ly link for e-book. This could be because the Bit.ly service is down – if the problems persists please let us know.
- **INVALID\_NUM\_PAGES** – You can only set a value of “1” or “2” for the **numPages** parameter to the **getBookreader** call. This error means you set some other value.

## listPublications

Method: **GET**

Returns: JSON array of objects representing publications.

Example:

<http://bookleteer.com/api/listPublications?key=795c8b3b319b90b3422ce2885000c540ab47f5b7>

## getPublicationPDF

Method: **GET**

Parameters: **id** – publication ID – *required*  
**pageSize** – one of *a4, a3, us, ledger* – *required*

Returns: a PDF file of the publication

Errors: **PUBLICATION\_NOT\_FOUND**  
**PAGE\_SIZE\_NOT\_FOUND**

**NOTE:** The parameter **key** is optional

## getEBookHTML

Method: **GET**

Parameters: **id** – publication ID – *required*

Returns: HTML associated with eBook, or JSON error. Use the HTTP content type to determine whether it's a JSON error or HTML. The HTML fragment is UTF-8 encoded.

Errors: **PUBLICATION\_NOT\_FOUND**  
**WRONG\_PUBLICATION\_TYPE**

## getBookreader

Method: **GET**

Parameters: **id** – publication ID – *required*

**numPages** – *required* – must be either “1” or “2”. This determines whether the embed code shows one or two pages.

**page** – *optional* – page number of page to show in embed code (first page is page 1, not page 0)

**NOTE:** Parameter **key** is optional.

Returns: JSON object with the following fields:

**url** – shortened (bit.ly) URL to Bookreader page for this eBook

**qr** – shortened (bit.ly) URL to QR code image (should be used inside `<img>`, in the `src` attribute).

**embedCode** – an `<iframe>` HTML fragment – copy & paste this into HTML page to embed a Bookreader widget displaying this eBook.

Errors: **PUBLICATION\_NOT\_FOUND**

**WRONG\_PUBLICATION\_TYPE**

**INVALID\_NUM\_PAGES**

**BITLY\_LINK\_ERROR**

## listImages

Method: **GET**

Parameters: *none*

Returns: JSON array of objects, each of which details info about an image.

### Description

This returns a list of image objects. An “image” in this context can mean 3 different things:

- An external URL
- An uploaded bitmap image (e.g. JPEG)
- An uploaded PDF

Note the following:

- An external URL to an image is just a plain URL, e.g. <http://foo.com/bar/house.jpg>
- For both uploaded bitmap images, and external URL images, the image can be used in the following parts of a publication:
  - logo
  - cover image
  - 2<sup>nd</sup> logo
  - StoryCube image
- An uploaded PDF can be used as the basis for either a StoryCube or eBook publication.
- For both uploaded bitmap images and uploaded PDFs, bookleeter.com caches “thumb” and “preview” sizes of the images, which you can fetch (preview size images are larger than thumb size).
- **NOTE:** The preview and thumb images are always JPEGs, irrespective of the original image type (even for PDFs).
- **NOTE:** Publications based on HTML content can reference external images or embed images. These images are *not* returned in this call.

Each JSON object returned can have the following fields. Note that some fields are only present in the

JSON object if they make sense for that object, e.g. if the image is not a PDF, then the numPages field doesn't exist in the returned JSON object. Fields can also be missing if they contain “null” values.

- **id** – use this (along with the **key** parameter) to get the image
- **width** – in pixels for bitmap images, or points (1/72 of an inch) for a PDF. If the latter then floating point values are cast to an integer.
- **height** – as above.
- **createdDateTime** – in format “YYYY-MM-DD HH:MM:SS”
- **pubs** – an array of publications IDs. It lists any publication ID which uses the image anywhere in the publication. The field is missing if the array is empty. If this field isn't present it means the image isn't used in any publication.
- **logos** – an array of publication IDs in which this image is used as the logo, or 2<sup>nd</sup> logo (field missing if array is empty)
- **covers** – as for **logos** but for the cover image
- **cubes** – as for **logos** but for the StoryCube images
- **uploadedPDFs** – as for **logos** but for publications which use this PDF as the source PDF for the publication (either eBook or StoryCube).
- **externalURL** – this field is only set if the image is external (not stored on bookleteer.com)

The following fields are only set for uploaded bitmap or PDF images (i.e. not external URLs):

- **fileName** – original file name when file was uploaded
- **MIMEType** – MIME type of uploaded file e.g. “image/gif”
- **numPages** – this is only set for PDF files
- **fullFileSize** – file size in bytes of the original uploaded file
- **previewFileSize** – byte size of preview image file
- **thumbFileSize** - byte size of thumb image file

## getImage

Method: **GET**

Parameters: **id** – image ID – *required*

Returns: Returns same JSON data as **listImages** but only for a single image

Errors: **IMAGE\_NOT\_FOUND**

## getImageFile

Method: **GET**

Parameters: **id** – image ID – *required*

**size** – one of *full*, *preview*, *thumb* – *optional* – default is 'thumb'

Returns: Image, or error in JSON format (use content type to tell which)

Errors: **IMAGE\_NOT\_FOUND**

**IMAGE\_SIZE\_NOT\_SUPPORTED**

**CANNOT\_GET\_EXTERNAL**

**IMAGE\_FILE\_NOT\_FOUND**

## listDesigns

Method: **GET**

Returns: JSON array of design objects. Each object has the following fields:

- **id** – unique numeric id that identifies design
- **name** – design name
- **forEbooks** – “1” if this design can be used to create eBooks, “0” otherwise
- **forStoryCubes** – “1” if this design can be used to create StoryCubes, “0” otherwise

## createImageFromURL

Method: **POST**

Parameters: **url** – url of image (must be a bitmap image type) – *required*

Returns: JSON object representing image, or error

Errors: **CURL\_ERROR**

**HTTP\_ERROR**

## createImageFromUpload

Method: **POST**

Parameters: **imageFile** – the file must be a PDF or a bitmap image type – *required*

Returns: JSON object representing image, or error

Errors:        **FILE\_UPLOAD\_ERROR**  
**NOT\_IMAGE**

### **Description**

The `imageFile` parameter has to be an actual file, not the name or ID of a file. For example, if you're using PHP and Curl to make this call to upload a local file called “flowers.pdf”, you'll need to prefix the file name with the '@' character, e.g. '@flowers.pdf' *not* 'flowers.pdf' (see PHP Curl documentation for details).

Here's an example of how you can upload a PDF file and then use it to create an EBook.

```
define('API_URI', 'http://bookleteer.com/api/');
define('API_KEY', 'alc0a0b42bfc963f47d80c0ff1e9293e692438e2');
define('OK', 'OK');

function makeCall(&$params, $callName) {
    $url = API_URI . $callName;
    // Setup call
    $h = curl_init($url);
    curl_setopt($h, CURLOPT_POST, true);
    curl_setopt($h, CURLOPT_POSTFIELDS, $params);
    // Execute and get call contents
    ob_start();
    curl_exec($h);
    $json = ob_get_contents();
    ob_end_clean();
    // Decode JSON
    $obj = json_decode($json);
    curl_close($h);

    return $obj;
}

// Upload source PDF
$params = array('key' => API_KEY, 'imageFile' => '@flowers.pdf');
$ret = makeCall($params, 'createImageFromUpload');
assert($ret->result == OK);

// Create EBook using newly uploaded PDF
$pdfId = $ret->payload->id;
$params = array('key' => API_KEY, 'pdfId' => $pdfId);
$ret = makeCall($params, 'editEBook');
assert($ret->result == OK);
```

### **deleteImage**

Method:        **POST**

Parameters:   **id** – image to delete

Returns:        JSON object representing error or **ok**

Errors:        **IMAGE\_NOT\_FOUND**  
**IMAGE\_IN\_USE**

### **editEBook**

Method:        **POST**

Parameters:   **id** – *optional* – publication id: if missing then create new EBook



**title** – *optional*

**author** – *optional*

**coverId** – *optional* – to remove existing image id set to empty string

**logoId** – *optional* – to remove existing image id set to empty string

**logo2Id** – *optional* – to remove existing image id set to empty string

**colophon** – *optional*

**date** – *optional*

**designId** – *optional* – for new publications defaults to “A4-US Basic”

**writingDirection** – *optional* – either “ltr” or “rtl”. New publications default to “ltr”

**format** – *optional* – either “book” or “classic”. New publications default to “classic”

**orientation** – *optional* – either “portrait” or “landscape”. New publications default to “portrait”

**pdfId** – *optional* – image id of uploaded PDF file used as content of publication. Takes precedence over html if present. Set to empty string to remove

**html** – *optional* – html content of publications (see description on how to embed images).

**isPublic** – *optional* – “0” (no) or “1” (yes). New publications default to “0”. This corresponds to the “Share with other members” check box.

**fileName** – *optional* – name used to form download name

**generatePDFs** – *optional* – “0” (no) or “1” (yes). New publications default to “0” but existing publications default to “0”. When set to “1”, the call will generate PDFs.

**bookreader** – *optional* – “0” (no) or “1” (yes). If set to “1”, other users will be able to view this publication using the Bookreader interface. This corresponds exactly to the “Share online with Bookreader” check box in the web UI.

- **NOTE 1:** The short URL to the Bookreader of this e-Book is created even if this parameter is set to “0” (though it's only created once for any given e-Book – the same URL is used throughout the life of the e-Book, no matter how many times you edit it).
- **NOTE 2:** Bookleteer lets you create a Bookreader link to an e-Book even if you haven't generated a PDF for it. To ensure that the Bookreader link works, make sure a PDF has been generated.

**authorBio** – *optional* – set author bio for the Bookreader (500 chars max)

**summary** – *optional* – set summary for the Bookreader (1000 chars max)

**publisherName** – *optional* – set publisher name for the Bookreader (100 chars max)

**copyright** – *optional* – set copyright text for the Bookreader (100 chars max)

Returns: JSON object representing publication, or error

Errors: **PUBLICATION\_NOT\_FOUND**  
**BAD\_DATE\_FORMAT**  
**BAD\_DESIGN\_ID**  
**BAD\_ORIENTATION**  
**IMAGE\_NOT\_FOUND**  
**WRONG\_IMAGE\_TYPE**  
**BAD\_WRITING\_DIRECTION**  
**BAD\_FORMAT**  
**PDF\_GENERATION\_ERROR**  
**BITLY\_LINK\_ERROR**

## Description

**editEBook** (along with **editStoryCube**) is the Swiss Army Knife of the API calls. You use this call to create/edit eBooks.

Note that all the parameters listed above are optional. If a parameter is missing in a call, and you're editing an existing publication, then the matching field is left alone, e.g. if you don't specify the **title** parameter then the existing publication's title is left unedited.

The **html** parameter should only contain a fragment in HTML, not a whole page. In particular, it should only contain the contents you would find within the `<body>` element (excluding the `<body>` tag itself). The **html** has to be encoded in UTF-8. To embed images, use the **data URI scheme** ([http://en.wikipedia.org/wiki/Data\\_URI\\_scheme](http://en.wikipedia.org/wiki/Data_URI_scheme)), e.g. here is an example HTML fragment with an embedded image:

```

```

```
<p>CND campaigns non-violently to rid the world of nuclear weapons and other weapons of mass destruction
and to create genuine security for future generations</p>
```

Here is some example code (in PHP) using the **editEBook** call:

```
define('API_URI', 'http://bookleteer.com/api/');
define('API_KEY', 'a1c0a0b42bfc963f47d80c0ff1e9293e692438e2');
define('OK', 'OK');

function makeCall(&$params, $callName) {
    $url = API_URI . $callName;
```

```
// Setup call
$h = curl_init($url);
curl_setopt($h, CURLOPT_POST, true);
curl_setopt($h, CURLOPT_POSTFIELDS, $params);
// Execute and get call contents
ob_start();
curl_exec($h);
$json = ob_get_contents();
ob_end_clean();
// Decode JSON
$obj = json_decode($json);
curl_close($h);

return $obj;
}

// Create new empty publication
$params = array('key' => API_KEY, 'generatePDFs' => '0');
$ret = makeCall($params, 'editEBook');
assert($ret->result == OK);

// Set some fields
$pubId = $ret->payload->id;
$params = array('key' => API_KEY, 'id' => $pubId, 'coverId' => 32,
               'title' => 'My Title', 'designId' => 2);
$ret = makeCall($params, 'editEBook');
assert($ret->result == OK);

// Finally, generate PDFs for eBook
$params = array('key' => API_KEY, 'id' => $pubId, 'generatePDFs' => '1');
$ret = makeCall($params, 'editEBook');
assert($ret->result == OK);
```

## editStoryCube

Method: **POST**

Parameters: **id** – *optional* – publication id: if missing then create new EBook

**title** – *optional*

**author** – *optional*

**cubeImageIdX** – *optional* – X is any number from 1-12.

**date** – *optional*

**designId** – *optional* – for new publications defaults to “A4 StoryCube”

**pdfId** – *optional* – image id of uploaded PDF file used as content of publication.

**isPublic** – *optional* – “0” (no) or “1” (yes). New publications default to “0”. This corresponds to the “Share with other members” check box.

**fileName** – *optional* – name used to form download name

**generatePDFs** – *optional* – “0” (no) or “1” (yes). New publications default to “0” but existing publications default to “0”. When set to “1”, the call will generate PDFs.

Returns: JSON object representing publication, or error

Errors: **PUBLICATION\_NOT\_FOUND**

**BAD\_DATE\_FORMAT**

**BAD\_DESIGN\_ID**

**IMAGE\_NOT\_FOUND**

**WRONG\_IMAGE\_TYPE**

**PDF\_GENERATION\_ERROR**

## **deletePublication**

Method: **POST**

Parameters: **id** – publication to delete

Returns: JSON object representing error or **ok**

Errors: **PUBLICATION\_NOT\_FOUND**