
epubaker Documentation

Release 1.1.0

Chen Meng

August 07, 2016

1	Add files	3
2	Spine	5
3	Metadata	7
4	Cover	9
5	Write it!	11
6	API Reference	13
6.1	Metas	15
6.2	tools	19
6.3	xl	20
7	Quick Start	23
8	Installing	25
9	Why Epubaker?	27
	Python Module Index	29

First of all, should have a book object:

epub3:

```
from epubaker import Epub3  
  
book = Epub3()
```

epub2:

```
from epubaker import Epub2  
  
book = Epub2()
```

Add files

Then, put a page into the book:

```
from epubaker import File

page1_path = 'p1.xhtml'
book.files[page1_path] = File(open('page1.xhtml', 'rb').read())
```

Spine

In print book, pages is just papers, when you open a book, you can see the pages. But epub book can stone lot of type of media, like audio, picture, fonts, and other things. Lot media is only be a part of a page, it not show up direct. So you have to let the book reader software knows what pages your want to show, and the show-pages order:

```
from epubaker import Joint
book.spine.append(Joint (page1_path))
```

That's it! That's minimum requirements of a useful book.

You may notice that we can't locating that page because we didn't make table of contents like a print book, do this to fix:

```
from epubaker import Section
book.toc.append(Section('Chapter I', page1_path))
```

Metadata

Now, we want reader know what's the title of this book, the identifier of this book and language of this book:

```
from epubaker.metas import Title, Identifier, Language

book.metadata.append(Title('simple epub book'))
book.metadata.append(Language('en'))
book.metadata.append(Identifier('any_string_different_from_other_identifier_of_other_book'))
```

epub3 need modified date:

```
from epubaker.tools import w3c_utc_date
from epubaker.metas import get_dcterm

book.metadata.append(get_dcterm('modified')(w3c_utc_date()))
```

Cover

Add a image first:

```
book.files['cover.png'] = File(open('cover.png', 'rb').read())
```

Let reader know which image is the cover:

epub3:

```
book.cover_path = 'cover.png'
```

epub2:

```
from epubaker metas import Cover
book.metadata.append(Cover('cover.png'))
```

If the reader or bookshelf didn't show the cover, you may want to make a xhtml page from the cover image, and put it as the first of the book pages:

```
cover_page_file = book.addons_make_image_page(image_path='cover.png')
book.files['cover.xhtml'] = cover_page_file
book.spine.insert(0, Joint('cover.xhtml'))
```

Write it!

```
book.write('simple_book.epub')
```

API Reference

`class epubaker.Epub3`

cover_image

Tag your cover image path as a cover

write (*filename*)

Write to file.

Parameters *filename* (*str*) – file name.

addons_make_toc_page ()

Some EPUB reader not supports nav hidden attribute, they just ignore sub section, not fold. So, this member function can make a toc page, with its little JS code, it can fold and unfold sections.

You must put the returned file to Epub3.files by yourself.

Returns xhtml page file

Return type *File*

addons_make_image_page (*image_path*, *cover_page_path=None*, *width=None*, *height=None*)

Make xhtml cover page contain the image you given.

You must put the returned file to *Epub2.files* by yourself

Parameters

- **image_path** – Image path in your Epub2.files
- **cover_page_path** – Use this to get relative path to the image path
- **width** – Image width, automatic recognition if None
- **height** – Image height, automatic recognition if None

Returns Cover xhtml page file.

Return type *File*

files

dict-like.

Store file path and *File* objects from *key* and *item*. Any file you want to package them into the book, you have to use this.

metadata

list-like.

Store metadata, such as author, publisher etc.

see *epubaker.metas*

spine

list-like.

“The spine defines the default reading order”

store *Joint* objects.

toc

list-like.

table of contents

store *Section* objects.

class *epubaker.Epub2*

write (*filename*)

Write to file.

Parameters *filename* (*str*) – file name.

addons_make_image_page (*image_path*, *cover_page_path=None*, *width=None*, *height=None*)

Make xhtml cover page contain the image you given.

You must put the returned file to *Epub2.files* by yourself

Parameters

- **image_path** – Image path in your Epub2.files
- **cover_page_path** – Use this to get relative path to the image path
- **width** – Image width, automatic recognition if None
- **height** – Image height, automatic recognition if None

Returns Cover xhtml page file.

Return type *File*

files

dict-like.

Store file path and *File* objects from *key* and *item*. Any file you want to package them into the book, you have to use this.

metadata

list-like.

Store metadata, such as author, publisher etc.

see *epubaker.metas*

spine

list-like.

“The spine defines the default reading order”

store *Joint* objects.

toc

list-like.

table of contents

store *Section* objects.

class `epubaker.File` (*binary*, *mime=None*, *fallback=None*)

Parameters

- **binary** (*bytes*) – binary data
- **mime** (*str*) – mime
- **fallback** (*str*) – file path

binary

as class parameter

class `epubaker.Section` (*title*, *href=None*)

Store title, href and sub *Section* objects.

Parameters

- **title** (*str*) – title of content.
- **href** (*str*) – html link to a file path in `Epub.files`, can have a bookmark. example:
text/a.html#hello

title

as class parameter

href

as class parameter

subs

list-like, store sub *Section* objects.

hidden_subs

bool: True for fold sub sections, False unfold.

some reader just don't show sub sections when this is True,

but I think it's mean FOLD sub sections and you can unfold it to show subs.

this i for epub3 nav only.

class `epubaker.Joint` (*path*, *linear=None*)

Parameters

- **path** (*str*) – file path, in `Epub.Files.keys()`
- **linear** (*bool*) – I don't know what is this mean. visit <http://idpf.org> to figure out by yourself.

path

as class parameter

6.1 Metas

6.1.1 Required Metadatas

class `epubaker.metas.Identifier` (*text*)

identifier

id
xml attributie: *id*

scheme
xml attribute: *opf:scheme*

class `epubaker.metas.Language` (*text*)
<https://tools.ietf.org/html/rfc5646>

example: en-US

id
xml attributie: *id*

class `epubaker.metas.Title` (*text*)
title of Book

alt_script
xml attribute: *opf:alt-script*

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

file_as
xml attribute: *opf:file-as*

id
xml attributie: *id*

lang
xml attribute: *xml:lang*

6.1.2 Option Metadatas

class `epubaker.metas.Contributor` (*text*)
contributor

alt_script
xml attribute: *opf:alt-script*

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

file_as
xml attribute: *opf:file-as*

id
xml attributie: *id*

lang
xml attribute: *xml:lang*

role
xml attribute: *opf:role*

class `epubaker.metas.Coverage` (*text*)
coverage

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

lang
xml attribute: *xml:lang*

class `epubaker.metas.Creator` (*text*)
creator

alt_script
xml attribute: *opf:alt-script*

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

file_as
xml attribute: *opf:file-as*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

role
xml attribute: *opf:role*

class `epubaker.metas.Date` (*text*)
date

id
xml attribute: *id*

class `epubaker.metas.Description` (*text*)
description

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

class `epubaker.metas.Format` (*text*)
format

id
xml attribute: *id*

class `epubaker.metas.Publisher` (*text*)
publisher

alt_script
xml attribute: *opf:alt-script*

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

file_as
xml attribute: *opf:file-as*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

class `epubaker.metas.Relation` (*text*)
relation

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

class `epubaker.metas.Rights` (*text*)
rights

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

class `epubaker.metas.Source` (*text*)
source

id
xml attribute: *id*

class `epubaker.metas.Subject` (*text*)
subject

authority
xml attribute: *authority*

dir
“ltr” (left-to-right) or “rtl” (right-to-left)
xml attribute: *dir*

id
xml attribute: *id*

lang
xml attribute: *xml:lang*

```
class epubaker.metas.Type(text)
    type
    id
        xml attributie: id
```

6.1.3 Only for Epub3

```
epubaker.metas.get_dcterm(name)
    get a term class by term name

class epubaker.metas.Meta3(property_, text)
    meta for Epub3.metadata

    alt_script
        xml attribute: opf:alt-script

    dir
        “ltr” (left-to-right) or “rtl” (right-to-left)
        xml attribute: dir

    file_as
        xml attribute: opf:file-as

    id
        xml attributie: id

    lang
        xml attribute: xml:lang

    property
        xml attribute: property

    scheme
        xml attribute: opf:scheme
```

6.1.4 Only for Epub2

```
class epubaker.metas.Meta2(name, content)
    meta for Epub2.metadata

class epubaker.metas.Cover(filepath)
    cover for Epub2.metadata
```

6.2 tools

```
epubaker.tools.relative_path(in_dir, to_file_path)
    if you got file with path “text/cover.xhtml” and it links to “image/cover.png”
```

Parameters

- **in_dir** – “text”
- **to_file_path** – “image/cover.png”

Returns “./image/cover.png”

```
epubaker.tools.identify_mime(binary)
```

Parameters `binary` – bytes html

Returns mime

`epubaker.tools.w3c_utc_date` (*date_time=None*)

Parameters `date_time` – instance of datetime, default is datetime.utcnow()

Returns like ‘CCYY-MM-DDThh:mm:ssZ’

6.3 xl

xml without mess.

This is a Python module to process XML.

6.3.1 Why I am coding this?

- lxml supports sub text node not clear enough.
- xml.etree.ElementTree doesn’t care XML namespaces.
- I want to learn XML.

`epubaker.xl.parse` (*xmlstr, debug=False*)

parse XML string to Xl object

Parameters

- `xmlstr` (*str*) –
- `debug` (*bool*) –

Returns object of *Xl*

Return type *Xl*

`epubaker.xl.clean_whitespace` (*element*)

Parameters `element` –

Returns A copy of the element, all whitespace characters have been stripped from the beginning and the end of the text node in the children and children’s children and so on. delete the text node If it is empty.

`epubaker.xl.pretty_insert` (*element, start_indent=0, step=4, dont_do_when_one_child=True*)

Modify the copy of the element, to make it looks more pretty and clear.

Parameters

- `element` (*Element*) –
- `start_indent` (*int*) –
- `step` (*int*) –
- `dont_do_when_one_child` (*bool*) –

Returns object of *Element*

`class epubaker.xl.Xl` (*header=None, doc_type=None, root=None*)

Parameters

- **header** (*Header*) –
- **doc_type** (*DocType*) –
- **root** (*Element*) –

header = None
object of *Header*

doc_type = None
object of *DocType*

root = None
object of *Element*

string ()
To xml string

class `epubaker.xml.Header` (*version=None, encoding=None, standalone=None*)
Handle XML header node

Parameters

- **version** (*str*) –
- **encoding** (*str*) –
- **standalone** (*bool*) –

class `epubaker.xml.DocType` (*doc_type_name, system_id, public_id*)
Handle XML doc type node

class `epubaker.xml.Element` (*tag=None, attributes=None, prefixes=None*)
Handle XML element node.

tag
tuple object of length 2.

First in the tuple is the url of the namespaces, the second is the xml element tag you know ordinarily.

string (*inherited_prefixes=None*)
to string, you may want to see *Xml.string*

Epubaker is a Python module to build EPUB 2 or 3 document from web files and related information.

Quick Start

it's a piece from User Guide:

```
from epubaker import Epub3, File, Joint

book = Epub3()

page1_path = '1.html'

book.files[page1_path] = File(open('page1.html', 'rb').read())

book.spine.append(Joint(page1_path))

book.write('my_book.epub')
```

Installing

```
pip install epubaker
```

or on Gentoo/Linux:

```
layman -a observer  
emerge -av epubaker
```

Why Epubaker?

- **New.** This module run under Python 2 and 3. It suporrts Epub 3, and Epub 2 too.
- **Clear.** epubaker doesn't modify the resource you were given. Files, metadata and other things are handled by different members of an Epub object.

e

`epubaker`, [13](#)
`epubaker.metas`, [15](#)
`epubaker.tools`, [19](#)
`epubaker.xl`, [20](#)

A

addons_make_image_page() (epubaker.Epub2 method), 14
addons_make_image_page() (epubaker.Epub3 method), 13
addons_make_toc_page() (epubaker.Epub3 method), 13
alt_script (epubaker.metas.Contributor attribute), 16
alt_script (epubaker.metas.Creator attribute), 17
alt_script (epubaker.metas.Meta3 attribute), 19
alt_script (epubaker.metas.Publisher attribute), 17
alt_script (epubaker.metas.Title attribute), 16
authority (epubaker.metas.Subject attribute), 18

B

binary (epubaker.File attribute), 15

C

clean_whitespace() (in module epubaker.xl), 20
Contributor (class in epubaker.metas), 16
Cover (class in epubaker.metas), 19
cover_image (epubaker.Epub3 attribute), 13
Coverage (class in epubaker.metas), 16
Creator (class in epubaker.metas), 17

D

Date (class in epubaker.metas), 17
Description (class in epubaker.metas), 17
dir (epubaker.metas.Contributor attribute), 16
dir (epubaker.metas.Coverage attribute), 16
dir (epubaker.metas.Creator attribute), 17
dir (epubaker.metas.Description attribute), 17
dir (epubaker.metas.Meta3 attribute), 19
dir (epubaker.metas.Publisher attribute), 17
dir (epubaker.metas.Relation attribute), 18
dir (epubaker.metas.Rights attribute), 18
dir (epubaker.metas.Subject attribute), 18
dir (epubaker.metas.Title attribute), 16
doc_type (epubaker.xl.Xl attribute), 21
DocType (class in epubaker.xl), 21

E

Element (class in epubaker.xl), 21
Epub2 (class in epubaker), 14
Epub3 (class in epubaker), 13
epubaker (module), 13
epubaker.metas (module), 15
epubaker.tools (module), 19
epubaker.xl (module), 20

F

File (class in epubaker), 15
file_as (epubaker.metas.Contributor attribute), 16
file_as (epubaker.metas.Creator attribute), 17
file_as (epubaker.metas.Meta3 attribute), 19
file_as (epubaker.metas.Publisher attribute), 18
file_as (epubaker.metas.Title attribute), 16
files (epubaker.Epub2 attribute), 14
files (epubaker.Epub3 attribute), 13
Format (class in epubaker.metas), 17

G

get_dcterm() (in module epubaker.metas), 19

H

Header (class in epubaker.xl), 21
header (epubaker.xl.Xl attribute), 21
hidden_subs (epubaker.Section attribute), 15
href (epubaker.Section attribute), 15

I

id (epubaker.metas.Contributor attribute), 16
id (epubaker.metas.Creator attribute), 17
id (epubaker.metas.Date attribute), 17
id (epubaker.metas.Description attribute), 17
id (epubaker.metas.Format attribute), 17
id (epubaker.metas.Identifier attribute), 15
id (epubaker.metas.Language attribute), 16
id (epubaker.metas.Meta3 attribute), 19
id (epubaker.metas.Publisher attribute), 18
id (epubaker.metas.Relation attribute), 18

id (epubaker.metas.Rights attribute), 18
id (epubaker.metas.Source attribute), 18
id (epubaker.metas.Subject attribute), 18
id (epubaker.metas.Title attribute), 16
id (epubaker.metas.Type attribute), 19
Identifier (class in epubaker.metas), 15
identify_mime() (in module epubaker.tools), 19

J

Joint (class in epubaker), 15

L

lang (epubaker.metas.Contributor attribute), 16
lang (epubaker.metas.Coverage attribute), 17
lang (epubaker.metas.Creator attribute), 17
lang (epubaker.metas.Description attribute), 17
lang (epubaker.metas.Meta3 attribute), 19
lang (epubaker.metas.Publisher attribute), 18
lang (epubaker.metas.Relation attribute), 18
lang (epubaker.metas.Rights attribute), 18
lang (epubaker.metas.Subject attribute), 18
lang (epubaker.metas.Title attribute), 16
Language (class in epubaker.metas), 16

M

Meta2 (class in epubaker.metas), 19
Meta3 (class in epubaker.metas), 19
metadata (epubaker.Epub2 attribute), 14
metadata (epubaker.Epub3 attribute), 13

P

parse() (in module epubaker.xl), 20
path (epubaker.Joint attribute), 15
pretty_insert() (in module epubaker.xl), 20
property (epubaker.metas.Meta3 attribute), 19
Publisher (class in epubaker.metas), 17

R

Relation (class in epubaker.metas), 18
relative_path() (in module epubaker.tools), 19
Rights (class in epubaker.metas), 18
role (epubaker.metas.Contributor attribute), 16
role (epubaker.metas.Creator attribute), 17
root (epubaker.xl.Xl attribute), 21

S

scheme (epubaker.metas.Identifier attribute), 16
scheme (epubaker.metas.Meta3 attribute), 19
Section (class in epubaker), 15
Source (class in epubaker.metas), 18
spine (epubaker.Epub2 attribute), 14
spine (epubaker.Epub3 attribute), 14
string() (epubaker.xl.Element method), 21

string() (epubaker.xl.Xl method), 21
Subject (class in epubaker.metas), 18
subs (epubaker.Section attribute), 15

T

tag (epubaker.xl.Element attribute), 21
Title (class in epubaker.metas), 16
title (epubaker.Section attribute), 15
toc (epubaker.Epub2 attribute), 14
toc (epubaker.Epub3 attribute), 14
Type (class in epubaker.metas), 18

W

w3c_utc_date() (in module epubaker.tools), 20
write() (epubaker.Epub2 method), 14
write() (epubaker.Epub3 method), 13

X

Xl (class in epubaker.xl), 20