pypub Documentation

Release 6

William Cember

June 07, 2016

Contents

Contents:

CHAPTER 1

pypub

Create epub's using python. Pypub is a python library to create epub files quickly without having to worry about the intracies of the epub specification.

1.1 Installation

The current release of pypub is available through pip:

```
$ pip install pypub
```

1.2 Copyright and License

CHAPTER 2

Tutorial - create an epub from reddit stories

Ready to get started? In this tutorial we will create an epub from the top stories currently on reddit.

Before you start, make sure the following is true

- Confirm that python 2 is installed on your machine and that you are using is. As of this writing, pypub is not compatible with python 3.
- Make sure the most recent version of pypub is installed
- In this tutorial, we are going to use the python package praw to access reddit's api. Download that too. While praw is necessary for this tutorial, you do not need it to use pypub.
- (Optional) If you want to read your ebook on kindle, you'll need to convert it to a mobi file. Download KindleGen to do this.

2.1 Find the stories you're interested

For this tutorial, we are going to grab the top stories from the subreddit r/TrueReddit. Pypub works with (most) websites, and the code contained here is easily generalizable to other content sources.

Let's begin by importing pypub as well as the other python libraries we're using for this tutorial

```
>>> import pypub
>>> import praw, os
```

Now let's get the current top ten stories from r/TrueReddit. The below code is necessary for this tutorial, but all we're returning here is a list of urls of stories we are interested in. Feel free to substitute here code to get whatever you are interested in (posts from your favorite blog, magazine articles, etc.)

```
>>> praw_object = praw.Reddit(user_agent='pypub/1.0')
>>> top_submission_list = praw_object.get_subreddit('TrueReddit').get_top(limit=10)
>>> top_submission_url_list = [submission.url for submission in top_submission_list]
```

top_submission_url_list is a python list of strings, where each string represents a url for one of the top ten stories from r/TrueReddit [1].

[1] If you are using a version of python earlier than 2.7.9, you may get SNIMissingWarning exception, which has to do with verifying HTTPS certificates. You should consider upgrading your version of python or following the instructions *here <http://urllib3.readthedocs.io/en/latest/security.html#snimissingwarning>*

2.2 Create the epub

Now that we have a list of url strings, we can use pypub to download the content and create an epub from the stories. First let's create an Epub object called epub. The only information we need to provide is the title of the ebook we'll be creating, which we will call 'TrueReddit - Top Stories'.

```
>>> epub = pypub.Epub('TrueReddit - Top Stories')
```

With the Epub object we just created, let's add a chapter to it for every story we in our url list.

```
>>> for url in top_submission_url_list:
...     try:
...         c = pypub.create_chapter_from_url(url)
...         epub.add_chapter(c)
...     except ValueError:
...         pass
```

Note in the above code that we try to create a chapter from every url, but don't if pypub raises a ValueError, which occurs if you try using pypub without an internet connection or if the url is invalid.

Finally, let's create our epub file. The below code saves it in the current working directory, but feel free to change that.

```
>>> epub.create_epub(os.getcwd())
```

At this point, in the directory you're working in, you should have a file *TrueReddit - Top Stories.epub*. This is your ebook with the top stories from r/TrueReddit and you should be able to read on your favorite device (unless that device is a kindle...see the next section).

2.3 (Optional) Convert to mobi

If you're like me, then you like reading things on your kindle (which is why I created pypub in the first place). Unfortunately, kindle uses it's own format for ebooks.

Luckily, Amazon provided a tool, KindleGen to convert epubs (and other formats) to mobi so they can be viewed on kindle. Once it's downloaded, convert *TrueReddit - Top Stories.epub* by going to the command prompt and entering kindlegen's full file path (excluding .exe) followed by "TrueReddit - Top Stories.epub".

> <full directory of kindlegen>kindlegen "TrueReddit - Top Stories.epub"

If the kindlegen executable is saved in the same directory as *TrueReddit - Top Stories.epub*, this can simply be entered as

> kindlegen "TrueReddit - Top Stories.epub"

CHAPTER 3

Developer Interface

3.1 Functions

pypub.**clean**(*input_string, tag_dictionary={'body': [], 'em': ['id', 'title'], 'head': [], 'blockquote': ['id'], 'img /': ['align', 'border', 'height', 'id', 'src', 'width'], 'h2': [], 'big': [], 'dd': ['id', 'title'], 'h1': [], 'h6': [], 'h4': [], 'h5': [], 'hr /': ['color', 'id', 'width'], 'ol': ['id'], 'br': ['id'], 'font': ['color', 'face', 'id', 'size'], 'strike': ['class', 'id'], 'a': ['href', 'id', 'name'], 'sub': ['id'], 'b': ['id'], 'span': ['bgcolor', 'title'], 'center': [], 'img': ['align', 'border', 'height', 'id', 'src', 'width'], 'ul': ['class', 'id'], 'i': ['class', 'id'], 'var': [], 'html': [], 'li': ['class', 'id', 'title'], 'p': ['align', 'id', 'title'], 's': ['id', 'style', 'title'], 'dfn': [], 'del': [], 'strong': ['class', 'id'], 'h3': [], 'small': ['id'], 'u': ['id'], 'div': ['align', 'id', 'bgcolor'], 'cite': [], 'sup': ['class', 'id']}*)

Sanitizes HTML. Tags not contained as keys in the tag_dictionary input are removed, and child nodes are recursively moved to parent of removed node. Attributes not contained as arguments in tag_dictionary are removed. Doctype is set to <!DOCTYPE html>.

Parameters

- **input_string** (*basestring*) – A (possibly unicode) string representing HTML.
- **tag_dictionary** (*Option[dict]*) – A dictionary with tags as keys and attributes as values. This operates as a whitelist–i.e. if a tag isn't contained, it will be removed. By default, this is set to use the supported tags and attributes for the Amazon Kindle, as found at https://kdp.amazon.com/help?topicId=A1JPUWCSD6F59O

Returns A (possibly unicode) string representing HTML.

Return type str

Raises `TypeError` – Raised if input_string isn't a unicode string or string.

3.2 Classes

class pypub.**Epub**(*title*, *creator='pypub'*, *language='en'*, *rights=''*, *publisher='pypub'*, *epub_dir=None*)

Class representing an epub. Add chapters to this and then output your ebook as an epub file.

Parameters

- **title** (*str*) – The title of the epub.
- **creator** (*Option[str]*) – The creator of your epub. By default this is pypub.
- **language** (*Option[str]*) – The language of your epub.

- **rights** (*Option[str]*) – The rights of your epub.
- **publisher** (*Option[str]*) – The publisher of your epub. By default this is pypub.

add_chapter(*c*)
Add a Chapter to your epub.

Parameters **c** (*Chapter*) – A Chapter object representing your chapter.

Raises `TypeError` – Raised if a Chapter object isn't supplied to this method.

create_epub(*output_directory*, *epub_name=None*)
Create an epub file from this object.

Parameters

- **output_directory** (*str*) – Directory to output the epub file to
- **epub_name** (*Option[str]*) – The file name of your epub. This should not contain .epub at the end. If this argument is not provided, defaults to the title of the epub.

class pypub.**Chapter**(*content*, *title*, *url=None*)
Class representing an ebook chapter. By and large this shouldn't be called directly but rather one should use the class ChapterFactor to instantiate a chapter.

Parameters

- **content** (*str*) – The content of the chapter. Should be formatted as xhtml.
- **title** (*str*) – The title of the chapter.
- **url** (*Option[str]*) – The url of the webpage where the chapter is from if applicable. By default this is None.

content
str

The content of the ebook chapter.

title
str

The title of the chapter.

url
str

The url of the webpage where the chapter is from if applicable.

html_title
str

Title string with special characters replaced with html-safe sequences

write(*file_name*)
Writes the chapter object to an xhtml file.

Parameters **file_name** (*str*) – The full name of the xhtml file to save to.

class pypub.**ChapterFactory**(*clean_function=<function clean at 0x7fbc8b02a5f0>*)
Used to create Chapter objects.Chapter objects can be created from urls, files, and strings.

Parameters **clean_function** (*Option[function]*) – A function used to sanitize raw html to be used in an epub. By default, this is the pypub.clean function.

create_chapter_from_file(*file_name*, *url=None*, *title=None*)

Creates a Chapter object from an html or xhtml file. Sanitizes the file's content using the clean_function method, and saves it as the content of the created chapter.

Parameters

- **file_name** (*string*) – The file_name containing the html or xhtml content of the created Chapter
- **url** (*Option[string]*) – A url to infer the title of the chapter from
- **title** (*Option[string]*) – The title of the created Chapter. By default, this is None, in which case the title will try to be inferred from the webpage at the url.

Returns A chapter object whose content is the given file and whose title is that provided or inferred from the url

Return type Chapter

create_chapter_from_string(*html_string*, *url=None*, *title=None*)

Creates a Chapter object from a string. Sanitizes the string using the clean_function method, and saves it as the content of the created chapter.

Parameters

- **html_string** (*string*) – The html or xhtml content of the created Chapter
- **url** (*Option[string]*) – A url to infer the title of the chapter from
- **title** (*Option[string]*) – The title of the created Chapter. By default, this is None, in which case the title will try to be inferred from the webpage at the url.

Returns A chapter object whose content is the given string and whose title is that provided or inferred from the url

Return type Chapter

create_chapter_from_url(*url*, *title=None*)

Creates a Chapter object from a url. Pulls the webpage from the given url, sanitizes it using the clean_function method, and saves it as the content of the created chapter. Basic webpage loaded before any javascript executed.

Parameters

- **url** (*string*) – The url to pull the content of the created Chapter from
- **title** (*Option[string]*) – The title of the created Chapter. By default, this is None, in which case the title will try to be inferred from the webpage at the url.

Returns A chapter object whose content is the webpage at the given url and whose title is that provided or inferred from the url

Return type Chapter

Raises `ValueError` – Raised if unable to connect to url supplied

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

A

C

E

H

T

U

W