pybase64 Documentation

Release 0.2.0

Matthieu Darbois

Aug 29, 2017

Contents:

This project is a wrapper on libbase64.

It aims to provide a fast base64 implementation for standard base64 encoding/decoding.

Installation

```
pip install pybase64
```

Usage

`pybase64` uses the same API as Python `base64` "modern interface" (introduced in Python 2.4) for an easy integration.

To get the fastest decoding, it is recommended to use the *b64decode()* and *validate=True* when possible.

```python
import pybase64

# Standard encoding
print(pybase64.standard_b64encode(b'>>>foo???'))        # b'Pj4+Zm9vPz8/'
print(pybase64.standard_b64decode(b'Pj4+Zm9vPz8/'))     # b'>>>foo???'
print(pybase64.urlsafe_b64encode(b'>>>foo???'))         # b'Pj4-Zm9vPz8_'
print(pybase64.urlsafe_b64decode(b'Pj4-Zm9vPz8_'))      # b'>>>foo???'
print(pybase64.b64encode(b'>>>foo???', altchars='_:'))  # b'Pj4_Zm9vPz8:'
print(pybase64.b64decode(b'Pj4_Zm9vPz8:', altchars='_:', validate=True)) # b'>>>foo???
↪'
```

Check *API* for more details.

A command-line tool is also provided. It has encode, decode and benchmark subcommands.

```
usage: pybase64 [-h] [-v] {benchmark,encode,decode} ...

pybase64 command-line tool.

positional arguments:
  {benchmark,encode,decode}
                        tool help
    benchmark           -h for usage
    encode              -h for usage
    decode              -h for usage

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
```

Benchmark

Running Python 3.6.0, Apple LLVM version 8.1.0 (clang-802.0.42), Mac OS X 10.12.6 on an Intel Core i7-4870HQ @ 2.50GHz

```
0.1.2 (C extension active)
bench: altchars=None, validate=False
pybase64._pybase64.b64encode:     3203.816 MB/s (13,271,472 bytes)
pybase64._pybase64.b64decode:      322.261 MB/s (13,271,472 bytes)
base64.b64encode:                  539.713 MB/s (13,271,472 bytes)
base64.b64decode:                  321.367 MB/s (13,271,472 bytes)
bench: altchars=None, validate=True
pybase64._pybase64.b64encode:     3119.150 MB/s (13,271,472 bytes)
pybase64._pybase64.b64decode:     4389.709 MB/s (13,271,472 bytes)
base64.b64encode:                  585.207 MB/s (13,271,472 bytes)
base64.b64decode:                  101.803 MB/s (13,271,472 bytes)
bench: altchars=b'-_', validate=False
pybase64._pybase64.b64encode:     2298.564 MB/s (13,271,472 bytes)
pybase64._pybase64.b64decode:      276.244 MB/s (13,271,472 bytes)
base64.b64encode:                  313.476 MB/s (13,271,472 bytes)
base64.b64decode:                  229.085 MB/s (13,271,472 bytes)
bench: altchars=b'-_', validate=True
pybase64._pybase64.b64encode:     2379.698 MB/s (13,271,472 bytes)
pybase64._pybase64.b64decode:     2862.796 MB/s (13,271,472 bytes)
base64.b64encode:                  315.344 MB/s (13,271,472 bytes)
base64.b64decode:                   91.621 MB/s (13,271,472 bytes)
```

3.1 API

3.1.1 Main API

`pybase64.`**`b64encode`**(*s*, *altchars=None*)
    Encode bytes using the standard Base64 alphabet.

Argument `s` is a bytes-like object to encode.

Optional `altchars` must be a byte string of length 2 which specifies an alternative alphabet for the '+' and '/' characters. This allows an application to e.g. generate url or filesystem safe Base64 strings.

The result is returned as a `bytes` object.

pybase64.**b64decode**(*s*, *altchars=None*, *validate=False*)
    Decode bytes encoded with the standard Base64 alphabet.

Argument `s` is a bytes-like object or ASCII string to decode.

Optional `altchars` must be a bytes-like object or ASCII string of length 2 which specifies the alternative alphabet used instead of the '+' and '/' characters.

If `validate` is `False` (the default), characters that are neither in the normal base-64 alphabet nor the alternative alphabet are discarded prior to the padding check. If `validate` is `True`, these non-alphabet characters in the input result in a `binascii.Error`.

The result is returned as a `bytes` object.

A `binascii.Error` is raised if `s` is incorrectly padded.

3.1.2 Helpers API

pybase64.**standard_b64encode**(*s*)
    Encode bytes using the standard Base64 alphabet.

Argument `s` is a bytes-like object to encode.

The result is returned as a `bytes` object.

pybase64.**standard_b64decode**(*s*)
    Decode bytes encoded with the standard Base64 alphabet.

Argument `s` is a bytes-like object or ASCII string to decode.

The result is returned as a `bytes` object.

A `binascii.Error` is raised if the input is incorrectly padded.

Characters that are not in the standard alphabet are discarded prior to the padding check.

pybase64.**urlsafe_b64encode**(*s*)
    Encode bytes using the URL- and filesystem-safe Base64 alphabet.

Argument `s` is a bytes-like object to encode.

The result is returned as a `bytes` object.

The alphabet uses '-' instead of '+' and '_' instead of '/'.

pybase64.**urlsafe_b64decode**(*s*)
    Decode bytes using the URL- and filesystem-safe Base64 alphabet.

Argument `s` is a bytes-like object or ASCII string to decode.

The result is returned as a `bytes` object.

A `binascii.Error` is raised if the input is incorrectly padded.

Characters that are not in the URL-safe base-64 alphabet, and are not a plus '+' or slash '/', are discarded prior to the padding check.

The alphabet uses '-' instead of '+' and '_' instead of '/'.

3.1.3 Information API

pybase64.**get_version**()
>   Returns pybase64 version as a `str` object.

>   The result reports if the C extension is used or not. e.g. *1.0.0 (C extension active)*

pybase64.**get_license_text**()
>   Returns pybase64 license information as a `str` object.

>   The result includes libbase64 license information as well.

3.2 License

3.2.1 pybase64

```
BSD 2-Clause License

Copyright (c) 2017, Matthieu Darbois
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

3.2.2 libbase64

```
Copyright (c) 2005-2007, Nick Galbreath
Copyright (c) 2013-2017, Alfred Klomp
Copyright (c) 2015-2017, Wojciech Mula
Copyright (c) 2016-2017, Matthieu Darbois
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
```

```
- Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Index

B

G

S

U