
b64uuid

Release 0.2

Liu Xue Yan

Sep 23, 2021

DOCS

1	Contents	3
2	Indices and tables	7
	Python Module Index	9
	Index	11

Release 0.2

Version 0.2.dev17+gc7c7427

Built-at Sep 23, 2021

CONTENTS

1.1 b64uuid

A small Python library and command-line tool to encode/decode UUID to/from a 22 characters shorter URL safe base64 string.

We can use it to make UUID string a little shorter.

1.1.1 Installation

- Installing from PyPI:

```
pip install b64uuid
```

- Installing from a local src tree:

```
pip install .
```

or

```
python setup.py install
```

Check <https://packaging.python.org/tutorials/installing-packages> for more details.

1.1.2 Command Line Usages

- Make a random short ID

```
$ b64uuid  
bxntPh4PSA6-0MDfBXMLhQ
```

- Short ID from UUID

```
$ b64uuid -u 2863a16d-b6ae-45a2-9d74-98d20377d56a  
KG0hbbauRaKddJjSA3fVag
```

- Short ID to UUID

```
$ b64uuid -s KG0hbbauRaKddJjSA3fVag  
2863a16d-b6ae-45a2-9d74-98d20377d56a
```

1.1.3 Library Usages

- Make a random short ID

```
>>> from b64uuid import B64UUID  
>>>  
>>> B64UUID().string  
'Ft01814aTwalxqDHMQoqTQ'
```

- Short ID from UUID

```
>>> from uuid import uuid1  
>>> from b64uuid import B64UUID  
>>>  
>>> uid = uuid1()  
>>> str(uid)  
'cb6e319c-d793-11ea-9619-1cb72cde3f7f'  
>>> bid = B64UUID(uid)  
>>> str(bid)  
'y24xnNeTEeqWGRy3LN4_fw'
```

- Short ID to UUID

```
>>> from uuid import uuid1  
>>> from b64uuid import B64UUID  
>>>  
>>> uid = uuid1()  
>>> uid.hex  
'95327416d79411ea96191cb72cde3f7f'  
>>> short_id = B64UUID(uid).string  
>>> short_id  
'1TJ0FteUEeqWGRy3LN4_fw'  
>>> B64UUID(short_id).uuid.hex  
'95327416d79411ea96191cb72cde3f7f'
```

1.2 AUTHORS

- Liu Xue Yan ([liu_xue_yan@foxmail.com](mailto.liu_xue_yan@foxmail.com))



1.3 Changelog

1.3.1 v0.1

- Date: 2020-08-06

1.4 b64uuid package

1.4.1 Submodules

b64uuid.b64uuid module

class b64uuid.b64uuid.B64UUID(*initial: Union[None, uuid.UUID, str, bytes, int] = None*)
 Bases: object

Class represents Base64 URL safe UUID

Parameters initial (None, uuid.UUID, str, bytes, int) – The parameter will be converted to `uuid.UUID`, and assigned to `uuid` property of the new created instance.

The way of converting to `uuid.UUID`:

- Case `None` (*default*) : Automatic generated by `uuid.uuid4()`
- Case `uuid.UUID` : No conversion
- **Case `str`** [It should be one of:]
 - An UUID string of 32 hexadecimal digits
 - A Base64 URL safe UUID string
- Case `bytes` : A string of 16 bytes in big-endian order
- Case `int` : A single 128-bit integer

property string: str

Base64 URL safe UUID of the instance

Returns An URL safe string represents the `UUID` value of the instance

Return type str

Note: The property is the same as converting to `str`:

```
from b64uuid import Base64UUID

bid = Base64UUID()
s1 = bid.string
s2 = str(bid)

assert s1 == s2 # equals!
```

property uuid: uuid.UUID

`UUID` of the instance

Returns Corresponding `stdlib` `UUID` object of the instance

Return type `uuid.UUID`

`b64uuid.b64uuid.b64id_to_uuid(s: str) → uuid.UUID`

Convert a Base64 URL safe ID string to an `uuid.UUID` object

Parameters `s (str)` – The UUID string in Base64 url safe style

Returns Corresponding *stdlib* `UUID` object

Return type `uuid.UUID`

`b64uuid.b64uuid.uuid_to_b64id(value: uuid.UUID) → str`

Convert an `uuid.UUID` object to a Base64 URL safe ID string

Parameters `value (uuid.UUID)` – The UUID to convert

Returns Base64 URL safe ID string

Return type `str`

b64uuid.version module

1.4.2 Module contents

**CHAPTER
TWO**

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

`b64uuid`, [6](#)
`b64uuid.b64uuid`, [5](#)
`b64uuid.version`, [6](#)

INDEX

B

b64id_to_uuid() (*in module b64uuid.b64uuid*), 6
b64uuid
 module, 6
B64UUID (*class in b64uuid.b64uuid*), 5
b64uuid.b64uuid
 module, 5
b64uuid.version
 module, 6

M

module
 b64uuid, 6
 b64uuid.b64uuid, 5
 b64uuid.version, 6

S

string (*b64uuid.b64uuid.B64UUID property*), 5

U

uuid (*b64uuid.b64uuid.B64UUID property*), 5
uuid_to_b64id() (*in module b64uuid.b64uuid*), 6