
hackr Documentation

Release 0.0.2

Ashwini Purohit

Dec 26, 2018

1	hackr: Unicorn library for Hackathons	1
1.1	Features	1
1.2	Installation	2
1.3	Usage	2
1.4	Contribution Guidelines	4

hackr: Unicorn library for Hackathons

An All-in-one library, safe for hackathon consumption.

You can perform every task using a single library.

Table of Contents

- *Welcome to hackr's documentation!*
- *hackr: Unicorn library for Hackathons*
 - *Features*
 - *Installation*
 - *Usage*
 - *Contribution Guidelines*

1.1 Features

hackr currently supports the following features:

- Generate random names
- Generate random address
- Generate random dates
- Generate random digits
- Generate random characters
- Generate QRCode
- Scrape any webpage

- Send web requests (GET, POST, PUT)
- Parse data in JSON
- Send Emails

hackr officially supports Python 2.7 currently.

1.2 Installation

```
pip install hackr
```

1.3 Usage

Generate Random names

```
>>> import hackr
>>> a = hackr.generator.names(5)
>>> print a
[u'Mr. Thomas Wolf MD', u'Valerie Turner', u'Maria Knight', u'Raymond Shelton', u
↪ 'Erica Glenn']
```

Generate Random dates

```
>>> import hackr
>>> a= hackr.generator.dates(5,1995,2017)
>>> print a
['1998-08-17 00:00:00', '2007-03-25 00:00:00', '2003-01-27 00:00:00', '2014-10-13_
↪ 00:00:00', '1995-02-10 00:00:00']
```

Generate Random json data

```
>>> import hackr
>>> a = hackr.object_generator.generate_json(4, name='names', address='address')
>>> print a
'[{ "name": "Jeffery Russell", "address": "USNV Bradley\\nFPO AA 49483-3369"}, {"name
↪ ": "Caitlin Wong", "address": "4622 Richard Summit Apt. 325\\nHollow
aystad, OH 88464"}, {"name": "Adrian Pugh", "address": "977 Hill Meadows Suite_
↪ 944\\nVictoriaton, PR 58653-2191"}, {"name": "Christopher Schaefer", "add
ress": "62215 Charles Cape Apt. 039\\nHaleymouth, ND 82518-8938"}]'
```

Generate QRCode

```
>>> import hackr
>>> # default save to current path
>>> img = hackr.image.qrcode("https://github.com/pytorn/hackr")
>>> # or manually assign dest path
>>> img = hackr.image.qrcode("https://github.com/pytorn/hackr", dest_path="/tmp/hackr_
↪ qrcode.png")
```

Scrape a webpage

```
>>> import hackr
>>> #To get the response as JSON(if the url returns a JSON response, otherwise an
↳exception is returned)
>>> json_response = hackr.web.scrape("https://api.github.com/users/ashwini0529", type=
↳"json")
>>> #To get the response as XML
>>> xml_response = hackr.web.scrape("https://api.github.com/users/ashwini0529", type=
↳"xml")
>>> #To scrape all images of a webpage(although the function will have limitations in
↳case of dynamically generated content)
>>> scrape("https://github.com/", type="json", images=True) # All the images from the
↳webpage will be saved in a folder named images inside a folder named as the URL.
```

Make Web Requests

```
>>> import hackr
>>> #If you want the response as a string
>>> a= hackr.web.request("http://httpbin.org/post", method="post", params={'a':'b'})
>>> If you want the response as JSON
>>> a= hackr.web.request("http://httpbin.org/post", method="post", params={'a':'b'},
↳type="JSON")
```

Send Emails

```
>>> import hackr
>>> #Enter your GMAIL email address and password in the parameters email, and
↳password.
>>> hackr.actions.email("Hey there", email="foo@bar.com", password="foobar", to=
↳"bar@foo.com", subject="Foo Bar")
```

IP Tools

```
>>> import hackr
>>> hackr.iptools.getLiveIP()
'***.***.***.***'
>>> hackr.iptools.getPrivateIP()
'192.168.1.8'
```

Cryptography

```
>>> import hackr
>>> e = hackr.crypto.Encrypt('hackr')
>>> e.sha1()
'0cbb7cc60b77fe81355c3b116837a5e50b747311'
>>> d = hackr.crypto.Decrypt('0cbb7cc60b77fe81355c3b116837a5e50b747311','dict.txt')
>>> d.sha1()
hackr
```

Cryptocurrencies

```
>>> import hackr
>>> hackr.crypto.Currency(1).convert('btc','eth')
32.16
>>> c = hackr.crypto.Currency(1)
>>> c.convert('btc','eth')
32.16
>>> c.coin = 'btc'
```

(continues on next page)

(continued from previous page)

```
>>> c.to('eth')
32.16
>>> c.value = 2
>>> c.to('eth')
64.32
```

1.4 Contribution Guidelines

- Fork the repository
- Find an issue or create one
- Create a branch(we prefer to name it patch)
- Inform everyone that you're working on the issue
- Send a pull request with proper explanation of what you did
- Wait for getting it reviewed.
- We'd be glad to merge your PR

Our Awesome [Contributors](#)

Made with [by: PyTorn](#)