FauxFactory Documentation

Release 3.0.0

Og Maciel

Jun 07, 2018

Contents

FauxFactory generates random data for your automated tests easily!

There are times when you're writing tests for your application when you need to pass random, non-specific data to the areas you are testing. For these scenarios when all you need is a random string, numbers, dates, times, email address, IP, etc, then FauxFactory can help!

CHAPTER 1

Installation

FauxFactory is available in PyPi and can be installed using pip:

```
$ pip install fauxfactory
```

You can install FauxFactory by downloading the latest version of the source code:

```
$ git clone git@github.com:omaciel/fauxfactory.git
$ cd fauxfactory
$ python setup.py build install
```

CHAPTER 2

Usage

Need a 15 character string for one of your tests?

```
>>> string = fauxfactory.gen_string('alphanumeric', 15)
>>> string.isalnum()
True
>>> len(string)
15
```

Need a 5 character numeric string?

```
>>> string = fauxfactory.gen_string('numeric', 5)
>>> string.isnumeric()
True
>>> len(string)
5
```

Now, let's say you need a random date:

```
>>> import datetime
>>> isinstance(fauxfactory.gen_date(), datetime.date)
True
>>> isinstance(fauxfactory.gen_datetime(), datetime.datetime)
True
```

Or a fake email with your company domain:

```
>>> email = fauxfactory.gen_email(domain='mycompany')
>>> '@mycompany' in email
True
```

Simple, right?

CHAPTER 3

Validation

All string functions allow validation of inputs using 3 parameters:

1. validator: a callable or str with regex returning boolean signaling if random data is valid or not.
2. tries: maximum number of times random data will be generated after failing validation. If the limit is reached "default" parameter will be returned.
3. default: value to be returned if validation fails a "tries" number of times.

Example using callable:

```
>>> def start_a(value):
...     return value[0] == 'a'
>>> email = fauxfactory.gen_email(validator=start_a, default = 'a@b.c')
>>> email[0] == 'a'
True
```

Example using regex:

```
>>> n = fauxfactory.gen_string(
... 'numeric', validator='[^0].*', default = '2')
>>> n != '0'
True
```

Example using tries and default:

```
>>> def always_false(value):
...     print('Executed')
...     return False
>>> fauxfactory.gen_alpha(
... validator=always_false, default = 'default value', tries=1)
Executed
'default value'
>>> fauxfactory.gen_alpha(
... validator=always_false, default = 'default value 2', tries=3)
Executed
```

(continues on next page)

(continued from previous page)

```
Executed
Executed
'default value 2'
```

CHAPTER 4

API

For a full list of available methods, see the *API documentation*.

4.1 API Documentation

This is a complete list of available methods, along with details about them.

4.1.1 fauxfactory

Generate random data for your tests.

4.1.2 fauxfactory.factories.booleans

Method for generating random boolean values.

fauxfactory.factories.booleans.**gen_boolean**()
Return a random Boolean value.

Returns A random Boolean value.

Return type bool

4.1.3 fauxfactory.factories.choices

Module to keep methods related to selecting values.

fauxfactory.factories.choices.**gen_choice**(*choices*)
Return a random choice from the available choices.

Parameters **choices** (*list*) – List of choices from which select a random value.

Raises ValueError if choices is None or not Iterable or a dict.

Returns A random element from `choices`.

`fauxfactory.factories.choices.`**`gen_uuid`**`()`
Generate a UUID string (universally unique identifiers).

Returns Returns a string representation for a UUID.

Return type str

4.1.4 fauxfactory.constants

Constants used by *fauxfactory*.

`fauxfactory.constants.`**`VALID_NETMASKS`**
A tuple of netmasks. The tuple index corresponds to a CIDR value. For example, a CIDR of "/1" corresponds to *VALID_NETMASKS[1]*.

4.1.5 fauxfactory.factories.dates

Methods related to generating date/time related values.

`fauxfactory.factories.dates.`**`gen_date`**`(`*min_date=None*, *max_date=None*`)`
Return a random date value.

Parameters

- **`min_date`** – A valid `datetime.date` object.
- **`max_date`** – A valid `datetime.date` object.

Raises `ValueError` if arguments are not valid `datetime.date` objects.

Returns Random `datetime.date` object.

`fauxfactory.factories.dates.`**`gen_datetime`**`(`*min_date=None*, *max_date=None*`)`
Return a random datetime value.

Parameters

- **`min_date`** – A valid `datetime.datetime` object.
- **`max_date`** – A valid `datetime.datetime` object.

Raises `ValueError` if arguments are not valid `datetime.datetime` objects.

Returns Random `datetime.datetime` object.

`fauxfactory.factories.dates.`**`gen_time`**`()`
Generate a random time.

Returns A random `datetime.time` object.

4.1.6 fauxfactory.helpers

Collection of helper methods and functions.

`class` `fauxfactory.helpers.`**`UnicodePlane`**`(`*min*, *max*`)`

`max`
Alias for field number 1

min
Alias for field number 0

fauxfactory.helpers.**base_repr**(*number*, *base*)
Return the base representation of a decimal number.

As shared here: https://stackoverflow.com/a/2267446

Conversion steps:

1. Divide the number by the base
2. Get the integer quotient for the next iteration
3. Get the remainder for the hex digit
4. Repeat the steps until quotient is equal to zero

Parameters

- **number** – (int) The decimal number to be converted.
- **base** – The base to convert.

Returns The base representation of <number>.

fauxfactory.helpers.**check_len**(*fnc*)
Validate generators requiring a *length* argument.

fauxfactory.helpers.**check_validation**(*fcn*)
Decorate functions requiring validation.

Simple decorator to validate values generated by function *fnc* according to parameters *validator*, *default* and *tries*.

Parameters **fcn** – function to be enhanced

Returns decorated function

fauxfactory.helpers.**is_positive_int**(*length*)
Check that *length* argument is an integer greater than zero.

Parameters **length** (*int*) – The desired length of the string

Raises *ValueError* if *length* is not an *int* or is less than 1.

fauxfactory.helpers.**unicode_letters_generator**(*smp=True*)
Generate unicode characters in the letters category.

Parameters **smp** (*bool*) – Include Supplementary Multilingual Plane (SMP) characters

Returns a generator which will generates all unicode letters available

4.1.7 fauxfactory.factories.internet

Methods related to generating internet related values.

fauxfactory.factories.internet.**gen_domain**(*name=None*, *subdomain=None*, *tlds=None*)
Generate a random domain name.

Parameters

- **name** (*str*) – Name for your host.
- **subdomain** (*str*) – Name for the subdomain.

- **tlds** (*str*) – Top Level Domain Server.

Returns A random domain name.

Return type str

fauxfactory.factories.internet.**gen_email**(*name=None*, *domain=None*, *tlds=None*)
Generate a random email address.

Parameters

- **name** (*str*) – Email name.
- **domain** (*str*) – Domain name.
- **tlds** (*str*) – Top Level Domain Server.

Returns An email address.

Return type str

fauxfactory.factories.internet.**gen_ipaddr**(*ip3=False*, *ipv6=False*, *prefix=()*)
Generate a random IP address.

You can also specify an IP address prefix if you are interested in local network address generation, etc.

Parameters

- **ip3** (*bool*) – Whether to generate a 3 or 4 group IP.
- **ipv6** (*bool*) – Whether to generate IPv6 or IPv4
- **prefix** (*list*) – A prefix to be used for an IP (e.g. [10, 0, 1]). It must be an iterable with strings or integers. Can be left unspecified or empty.

Returns An IP address.

Return type str

Raises `ValueError` if `prefix` would lead to no random fields at all. This means the length that triggers the `ValueError` is 4 for regular IPv4, 3 for IPv4 with ip3 and 8 for IPv6. It will be raised in any case the prefix length reaches or exceeds those values.

fauxfactory.factories.internet.**gen_mac**(*delimiter=':'*, *multicast=None*, *locally=None*)
Generate a random MAC address.

For more information about how unicast or multicast and globally unique and locally administered MAC addresses are generated check this link https://en.wikipedia.org/wiki/MAC_address.

Parameters

- **delimeter** (*str*) – Valid MAC delimeter (e.g ':', '-').
- **multicast** (*bool*) – Indicates if the generated MAC address should be unicast or multicast. If no value is provided a random one will be chosen.
- **locally** (*bool*) – Indicates if the generated MAC address should be globally unique or locally administered. If no value is provided a random one will be chosen.

Returns A random MAC address.

Return type str

fauxfactory.factories.internet.**gen_netmask**(*min_cidr=1*, *max_cidr=31*)
Generate a random valid netmask.

For more info: http://www.iplocation.net/tools/netmask.php

Parameters

- **min_cidr** (*int*) – Inferior CIDR limit
- **max_cidr** (*int*) – Superior CIDR limit

Returns The netmask is chosen from *fauxfactory.constants.VALID_NETMASKS* respecting the CIDR range

Return type str

Raises `ValueError` if `min_cidr` or `max_cidr` have an invalid value. For example, `max_cidr` cannot be 33.

`fauxfactory.factories.internet.`**`gen_url`**(*scheme=None*, *subdomain=None*, *tlds=None*)
Generate a random URL address.

Parameters

- **scheme** (*str*) – Either http, https or ftp.
- **subdomain** (*str*) – A valid subdmain
- **tlds** (*str*) – A qualified top level domain name (e.g. 'com', 'net')

Raises `ValueError` if arguments are not valid.

Returns A random URL address.

Return type str

4.1.8 fauxfactory.factories.numbers

Methods that generate random number values.

`fauxfactory.factories.numbers.`**`gen_integer`**(*min_value=None*, *max_value=None*)
Return a random integer value based on the current platform.

Parameters

- **min_value** (*int*) – The minimum allowed value.
- **max_value** (*int*) – The maximum allowed value.

Raises `ValueError` if arguments are not integers or if they are less or greater than the system's allowed range for integers.

Returns Returns a random integer value.

Return type int

`fauxfactory.factories.numbers.`**`gen_negative_integer`**()
Return a random negative integer based on the current platform.

Returns Returns a random negative integer value.

Return type int

`fauxfactory.factories.numbers.`**`gen_number`**(*min_value=None*, *max_value=None*, *base=10*)
Return a random number (with <base> representation).

Returns A random number with base of <base>.

Return type str

`fauxfactory.factories.numbers.`**`gen_positive_integer`**()
Return a random positive integer based on the current platform.

Returns A random positive integer value.

Return type int

4.1.9 fauxfactory.factories.strings

Collection of string generating functions.

fauxfactory.factories.strings.**gen_alpha**(*length=10*)
Return a random string made up of alpha characters.

Parameters **length** (*int*) – Length for random data.

Returns A random string made up of alpha characters.

Return type str

fauxfactory.factories.strings.**gen_alphanumeric**(*length=10*)
Return a random string made up of alpha and numeric characters.

Parameters **length** (*int*) – Length for random data.

Returns A random string made up of alpha and numeric characters.

Return type str

fauxfactory.factories.strings.**gen_cjk**(*length=10*)
Return a random string made up of CJK characters.

(Source: Wikipedia - CJK Unified Ideographs)

Parameters **length** (*int*) – Length for random data.

Returns A random string made up of CJK characters.

Return type str

fauxfactory.factories.strings.**gen_cyrillic**(*length=10*)
Return a random string made up of Cyrillic characters.

Parameters **length** (*int*) – Length for random data.

Returns A random string made up of Cyrillic characters.

Return type str

fauxfactory.factories.strings.**gen_html**(*length=10*, *include_tags=True*)
Return a random string made up of html characters.

Parameters **length** (*int*) – Length for random data.

Returns A random string made up of html characters.

Return type str

fauxfactory.factories.strings.**gen_iplum**(*words=None*, *paragraphs=None*)
Return a lorem ipsum string.

If no arguments are passed, then return the entire default lorem ipsum string.

Parameters

- **words** (*int*) – The number of words to return.
- **paragraphs** (*int*) – The number of paragraphs to return.

Raises `ValueError` if `words` is not a valid positive integer.

Returns A `lorem ipsum` string containing either the number of `words` or `paragraphs`, extending and wrapping around the text as needed to make sure that it has the specified length.

Return type str

`fauxfactory.factories.strings.`**`gen_latin1`**(*length=10*)

Return a random string made up of UTF-8 characters.

(Font: Wikipedia - Latin-1 Supplement Unicode Block)

Parameters **`length`** (*int*) – Length for random data.

Returns A random string made up of `Latin1` characters.

Return type str

`fauxfactory.factories.strings.`**`gen_numeric_string`**(*length=10*)

Return a random string made up of numbers.

Parameters **`length`** (*int*) – Length for random data.

Returns A random string made up of numbers.

Return type str

`fauxfactory.factories.strings.`**`gen_special`**(*length=10*)

Return a random special characters string.

Parameters **`length`** (*int*) – Length for random data.

Returns A random string made up of special characters.

Return type str

`fauxfactory.factories.strings.`**`gen_string`**(*str_type*, *length=None*, *validator=None*, *default=None*, *tries=10*)

A simple wrapper that calls other string generation methods.

Parameters

- **`str_type`** (*str*) – The type of string which should be generated.
- **`length`** (*int*) – The length of the generated string. Must be 1 or greater.
- **`validator`** – Function or regex (str). If a function it must receive one parameter and return True if value can be used and False of another value need to be generated. If str it will be used as regex to validate the generated value. Default is None which will not validate the value.
- **`tries`** – number of times validator must be called before returning *default*. Default is 10.
- **`default`** – If validator returns false a number of *tries* times, this value is returned instead. Must be defined if validator is not None

Raises `ValueError` if an invalid `str_type` is specified.

Returns A string.

Return type str

Valid values for `str_type` are as follows:

- alpha
- alphanumeric
- cjk
- cyrillic

- html
- latin1
- numeric
- utf8
- punctuation

fauxfactory.factories.strings.**gen_utf8**(*length=10*, *smp=True*)
Return a random string made up of UTF-8 letters characters.

Follows RFC 3629.

Parameters

- **length** (*int*) – Length for random data.
- **smp** (*bool*) – Include Supplementary Multilingual Plane (SMP) characters

Returns A random string made up of UTF-8 letters characters.

Return type str

4.1.10 fauxfactory.factories.systems

Collection of computer systems generating functions.

fauxfactory.factories.systems.**add_memory_info**(*count=None*)
Generate fake memory facts.

Parameters **count** (*int*) – The total amount of RAM for a system.

Returns A dictionary representing memory facts.

Return type dict

fauxfactory.factories.systems.**add_network_devices**()
Generate fake network device facts.

Returns A dictionary representing a Host's network devices.

Return type dict

fauxfactory.factories.systems.**add_operating_system**(*name=None*, *family=None*, *major=None*, *minor=None*)
Generate fake operating system facts.

Parameters

- **name** (*str*) – The name for an operating system.
- **family** (*str*) – The operating system family.
- **major** (*int*) – The major release of the operating system.
- **minor** (*int*) – The minor release of the operating system.

Returns A dictionary representing an Operating System.

Return type dict

fauxfactory.factories.systems.**add_partitions**(*extra_partitions=None*)
Generate fake partitions facts.

fauxfactory.factories.systems.**add_processor_info**(*count=None*)
Generate fake processor facts.

Parameters **count** (*int*) – Number of processors for a system.

Returns A dictionary containing fake processor facts.

Return type dict

fauxfactory.factories.systems.**gen_system_facts**(*name=None*)
Generate system facts.

See https://docs.puppet.com/facter/3.6/core_facts.html for more information.

Parameters **name** (*str*) – Name to be used as the system's hostname.

Returns A Dictionary representing a system's facts.

Return type dict

CHAPTER 5

Contribute

1. Fork the repository on GitHub and make some changes. Make sure to add yourself to AUTHORS.
2. Install the development requirements. `pip install -r requirements-optional.txt`.
3. Test your changes.
   (a) Run `make test-all` and make sure nothing has broken.
   (b) Run `coverage report --show-missing` to check for untested code.
   (c) Add tests to the `tests/` directory if appropriate.

   Repeat this cycle as needed.
4. Send a pull request and bug the maintainer until it gets merged and published. :)

Python Module Index

f

Index

U

V