

---

# **iam-profile-faker Documentation**

***Release 0.3.1***

**John Giannelos**

**Nov 16, 2018**



---

## Contents:

---

<b>1</b>	<b>iam-profile-faker</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Credits . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Stable release . . . . .	3
2.2	From sources . . . . .	3
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>iam_profile_faker</b>	<b>7</b>
4.1	iam_profile_faker package . . . . .	7
<b>5</b>	<b>Contributing</b>	<b>9</b>
5.1	Types of Contributions . . . . .	9
5.2	Get Started! . . . . .	10
5.3	Pull Request Guidelines . . . . .	11
5.4	Tips . . . . .	11
5.5	Deploying . . . . .	11
<b>6</b>	<b>Credits</b>	<b>13</b>
6.1	Development Lead . . . . .	13
6.2	Contributors . . . . .	13
<b>7</b>	<b>History</b>	<b>15</b>
7.1	0.3.0 (2018-07-19) . . . . .	15
7.2	0.2.0 (2018-07-13) . . . . .	15
7.3	0.1.0 (2018-07-10) . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



JSON generator based on IAM profile v2 schema

- Free software: Apache Software License 2.0
- Documentation: <https://iam-profile-faker.readthedocs.io>.

## 1.1 Features

- Create fake profile v2 objects
- Batch create fake profile v2 objects
- CLI to export profile v2 objects in JSON.

## 1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.



### 2.1 Stable release

To install iam-profile-faker, run this command in your terminal:

```
$ pip install iam_profile_faker
```

This is the preferred method to install iam-profile-faker, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for iam-profile-faker can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/mozilla-iam/iam-profile-faker
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/mozilla-iam/iam-profile-faker/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```





## CHAPTER 3

---

### Usage

---

To use iam-profile-faker in a project:

```
from iam_profile_faker.factory import V2ProfileFactory

factory = V2ProfileFactory()
# Generate a single object
factory.create()

# Generate a batch of 10 objects
factory.create_batch(10)

# Generate a single object serialized to JSON
factory.create(export_json=True)

# Generate a batch of 10 objects serialized to JSON
factory.create_batch(10, export_json=True)
```

To use iam-profile-faker as a CLI tool:

```
$ iam_profile_faker --help
Usage: iam_profile_faker [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  create      Create single IAM profile v2 object.
  create_batch Create batch IAM profile v2 objects.
```



## 4.1 iam\_profile\_faker package

### 4.1.1 Submodules

### 4.1.2 iam\_profile\_faker.cli module

Console script for iam\_profile\_faker.

### 4.1.3 iam\_profile\_faker.factory module

```
class iam_profile_faker.factory.IAMFaker (locale=None, hierarchy=None)
    Bases: object

    access_information ()
        Profile v2 access information faker.

    create ()
        Method to generate fake profile v2 objects.

    get_public_email_address ()

    hris ()
        Profile v2 HRIS faker

    identities ()
        Profile v2 identities faker.

    languages (**kwargs)

    login_method (**kwargs)

    metadata (display=['public', 'authenticated', 'vouched', 'ndaed', 'staff', 'private'],
               c12n='WORKGROUP CONFIDENTIAL')
        Generate field metadata
```

```
office_location()
    Profile v2 office location faker.

pgp_public_keys(**kwargs)

phone_numbers(**kwargs)

pronouns(**kwargs)

schema()
    Profile v2 schema faker.

signature()
    Generate field signature

ssh_public_keys(**kwargs)

staff_information(hris)
    Profile v2 staff information faker

uris(**kwargs)

user_id(**kwargs)

usernames(**kwargs)

class iam_profile_faker.factory.V2ProfileFactory
    Bases: object

    create(export_json=False)
        Generate fake profile v2 object.

    create_batch(count, export_json=False)
        Generate batch fake profile v2 objects.

iam_profile_faker.factory.create_random_hierarchy_iter()
    Generate hierarchy iterator with a random pattern

iam_profile_faker.factory.decorate_metadata_signature(display=['public',      'au-
    thenticated',      'vouched',
    'ndaed',      'staff',      'private'],
    c12n='WORKGROUP CON-
    FIDENTIAL')

iam_profile_faker.factory.wrap_metadata_signature(obj, value, display=['public',
    'authenticated',      'vouched',
    'ndaed',      'staff',      'private'],
    c12n='WORKGROUP      CONFI-
    DENTIAL')

    Wrap profile value with metadata/signature
```

#### 4.1.4 Module contents

Top-level package for iam-profile-faker.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 5.1 Types of Contributions

### 5.1.1 Report Bugs

Report bugs at <https://github.com/mozilla-iam/iam-profile-faker/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### 5.1.4 Write Documentation

iam-profile-faker could always use more documentation, whether as part of the official iam-profile-faker docs, in docstrings, or even on the web in blog posts, articles, and such.

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mozilla-iam/iam-profile-faker/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up *iam\_profile\_faker* for local development.

1. Fork the *iam\_profile\_faker* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/iam_profile_faker.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv iam_profile_faker
$ cd iam_profile_faker/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 iam_profile_faker tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check [https://travis-ci.org/mozilla-iam/iam-profile-faker/pull\\_requests](https://travis-ci.org/mozilla-iam/iam-profile-faker/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_iam_profile_faker
```

## 5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.





### 6.1 Development Lead

- John Giannelos <[johngiannelos@mozilla.com](mailto:johngiannelos@mozilla.com)>

### 6.2 Contributors

None yet. Why not be the first?



### 7.1 0.3.0 (2018-07-19)

- Fix factory to wrap values with metadata/signature
- Fix value/values key name in various attributes
- Add E2E to tests to validate objects against profile v2
- Expose a service that generates fake profiles as HTTP API

### 7.2 0.2.0 (2018-07-13)

- Add factory to generate fake iam profiles
- Add cli to generate fake iam profiles

### 7.3 0.1.0 (2018-07-10)

- Bootstrap repository



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### i

`iam_profile_faker`, 8  
`iam_profile_faker.cli`, 7  
`iam_profile_faker.factory`, 7





## A

`access_information()` (`iam_profile_faker.factory.IAMFaker` method), 7

## C

`create()` (`iam_profile_faker.factory.IAMFaker` method), 7  
`create()` (`iam_profile_faker.factory.V2ProfileFactory` method), 8  
`create_batch()` (`iam_profile_faker.factory.V2ProfileFactory` method), 8  
`create_random_hierarchy_iter()` (in module `iam_profile_faker.factory`), 8

## D

`decorate_metadata_signature()` (in module `iam_profile_faker.factory`), 8

## G

`get_public_email_address()`  
(`iam_profile_faker.factory.IAMFaker` method), 7

## H

`hris()` (`iam_profile_faker.factory.IAMFaker` method), 7

## I

`iam_profile_faker` (module), 8  
`iam_profile_faker.cli` (module), 7  
`iam_profile_faker.factory` (module), 7  
`IAMFaker` (class in `iam_profile_faker.factory`), 7  
`identities()` (`iam_profile_faker.factory.IAMFaker` method), 7

## L

`languages()` (`iam_profile_faker.factory.IAMFaker` method), 7  
`login_method()` (`iam_profile_faker.factory.IAMFaker` method), 7

## M

`metadata()` (`iam_profile_faker.factory.IAMFaker` method), 7

## O

`office_location()` (`iam_profile_faker.factory.IAMFaker` method), 7

## P

`pgp_public_keys()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`phone_numbers()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`pronouns()` (`iam_profile_faker.factory.IAMFaker` method), 8

## S

`schema()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`signature()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`ssh_public_keys()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`staff_information()` (`iam_profile_faker.factory.IAMFaker` method), 8

## U

`uris()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`user_id()` (`iam_profile_faker.factory.IAMFaker` method), 8  
`usernames()` (`iam_profile_faker.factory.IAMFaker` method), 8

## V

`V2ProfileFactory` (class in `iam_profile_faker.factory`), 8

## W

`wrap_metadata_signature()` (in module `iam_profile_faker.factory`), 8