

fakerservicio.py - proyecto_microservicio - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- PROYECTO_MICROSERVICIO
 - api
 - database
 - faker
 - projectoweb
 - fakerservicio.py
 - web

fakerservicio.py

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

lg@lg-virtual-machine:~/projecto_microservicio\$ python3 -m venv projectoweb

- lg@lg-virtual-machine:~/projecto_microservicio\$ ls
- api database faker projectoweb web
- lg@lg-virtual-machine:~/projecto_microservicio\$ cd faker/
- lg@lg-virtual-machine:~/projecto_microservicio/faker\$ python3 -m venv projectoweb
- lg@lg-virtual-machine:~/projecto_microservicio/faker\$

bash - faker

Edit Selection View Go Run Terminal Help

EXPLORER

- PROYECTO_MICROSERVICIO
 - api
 - database
 - faker
 - projectoweb
 - fakerservicio.py
 - web

fakerservicio.py

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

lg@lg-virtual-machine:~/projecto_microservicio\$ python3 -m venv projectoweb

- lg@lg-virtual-machine:~/projecto_microservicio\$ ls
- api database faker projectoweb web
- lg@lg-virtual-machine:~/projecto_microservicio\$ cd faker/
- lg@lg-virtual-machine:~/projecto_microservicio/faker\$ python3 -m venv projectoweb
- lg@lg-virtual-machine:~/projecto_microservicio/faker\$ source projectoweb/bin/activate
- (projectoweb) lg@lg-virtual-machine:~/projecto_microservicio/faker\$

bash - faker

```
fakerservicio.py - proyecto_microservicio - Visual Studio Code

Edit Selection View Go Run Terminal Help

EXPLORER
PROYECTO_MICROSERVICIO
  api
  database
  faker
  projectoweb
  + fakerservicio.py
  web

fakerservicio.py
1

TERMINAL
bash - faker
setuptools 59.6.0
• (projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip install flask
Collecting flask
Using cached Flask-2.2.3-py3-none-any.whl (101 kB)
Collecting Jinja2==3.0
Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting click>=8.0
Using cached click-8.1.3-py3-none-any.whl (96 kB)
```

```
(projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip freeze
click==8.1.3
Flask==2.2.3
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.2
Werkzeug==2.2.3
(projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$
```

<https://faker.readthedocs.io/en/master/>



maestro

Buscar documentos

Usando la clase Faker

Proveedores estándar

Proveedores de la comunidad

Proveedores localizados

Accesorios Pytest

Estilo de codificación

Escritura de documentación



Lea los documentos para empresas:
repositorios privados con vistas previas de solicitud de incorporación de cambios. **Comience su prueba gratis hoy.**

Anuncio de EthicalAds

¡Bienvenido a la documentación de Faker!

Faker es un paquete de Python que genera datos falsos para ti. Ya sea que necesite iniciar su base de datos, crear documentos XML atractivos, completar su persistencia para realizar pruebas de estrés o anonimizar los datos tomados de un servicio de producción, Faker es para usted.

Faker está fuertemente inspirado en [PHP Faker](#), [Perl Faker](#) y [Ruby Faker](#).

version

v17.6.0

Python Tests

failing

coverage

96%

license

MIT

Compatibilidad

A partir de la versión `4.0.0`, `Faker` se eliminó la compatibilidad con Python 2 y desde la versión `5.0.0` solo se admite Python 3.7 y superior. Si aún necesita la compatibilidad con Python 2, instale la versión `3.0.1` mientras tanto y considere actualizar su base de código para admitir Python 3 para que pueda disfrutar de las últimas funciones `Faker` que ofrece. Consulte los [documentos ampliados](#) para obtener más detalles, especialmente si está actualizando desde una versión `2.0.4` anterior, ya

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Using cached MarkupSafe-2.1.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.2.3 click-8.1.3 flask-2.2.3 itsdangerous-2.1.2
(projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip freeze
click==8.1.3
Flask==2.2.3
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.2
Werkzeug==2.2.3
(projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip install Faker
```

```
Werkzeug==2.2.3
(projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip install Faker
Collecting Faker
  Downloading Faker-17.6.0-py3-none-any.whl (1.7 MB)
    1.2/1.7 MB 333.7 kB/s e Select End of Line Sequence
```

```
EXPLORER
PROYECTO_MICROSERVICIO
  api
  database
  faker
    > projectoweb
    + fakerservicio.py 1
  > web

fakerservicio.py 1
Faker > fakerservicio.py > ...
3 app = Flask(__name__)
4 faker = Faker()
5
6 @app.route('/crear_registro')
7 def create_profiles():
8     try:
9         return {"message": "Creando Registro"}
10    except:
11        return {"Error en crear registro"}
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=3000, debug=True)]

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
• (projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ pip install Faker
Collecting Faker
  Downloading Faker-17.6.0-py3-none-any.whl (1.7 MB)
    1.7/1.7 MB 112.7 kB/s eta 0:00:00
Collecting python-dateutil>=2.4
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, python-dateutil, Faker
Successfully installed Faker-17.6.0 python-dateutil-2.8.2 six-1.16.0
○ (projectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$
```

File Edit Selection View Go Run Terminal Help

EXPLORER

- PROJECTO_MICROSERVICIO
 - api
 - database
 - faker
 - proyectoweb
 - fakerservicio.py
 - web

fakerservicio.py 1

```
3 app = Flask(__name__)
4 faker = Faker()
5
6 @app.route('/crear_registro')
7 def create_profiles():
8     try:
9         return {"message": "Creando Registro"}
10    except:
11        return {"Error en crear registro"}
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=3000, debug=True)
```

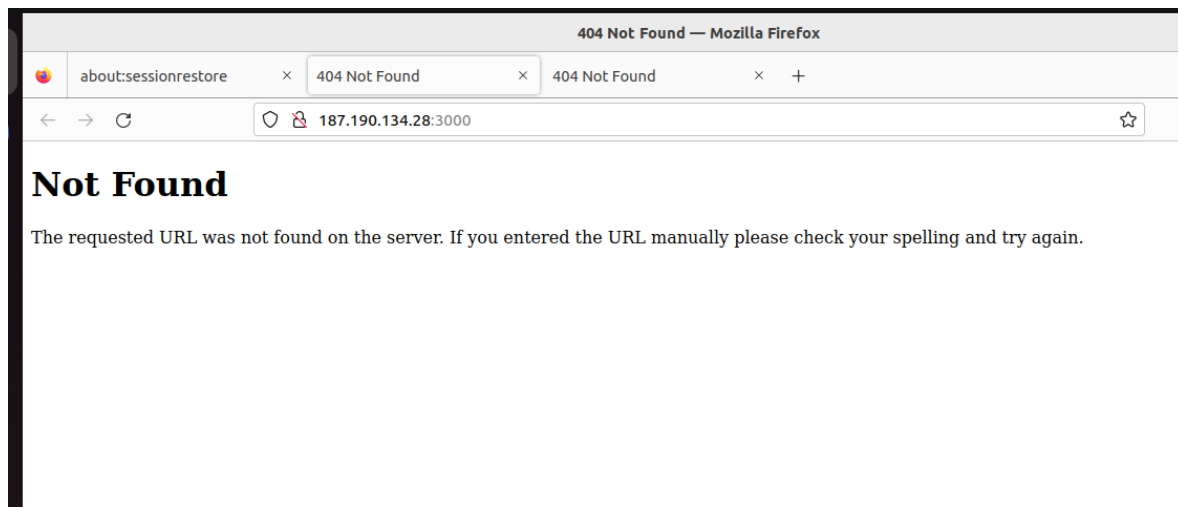
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

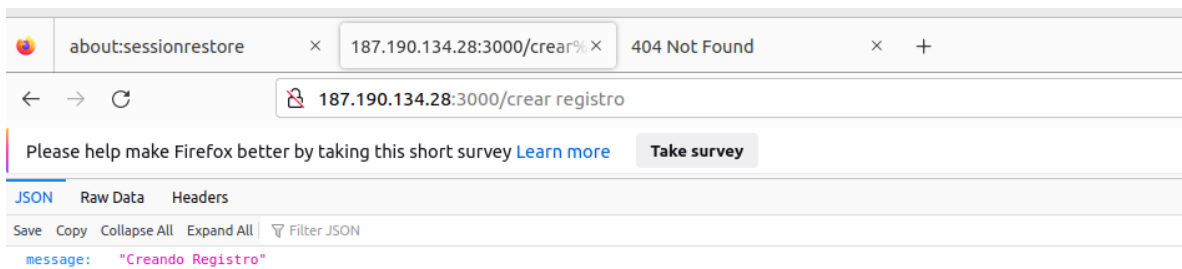
python3 - faker

fakerservicio.py proyectoweb

- (proyectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker\$ python3 fakerservicio.py
 - * Serving Flask app 'fakerservicio'
 - * Debug mode: on
 - WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 - * Running on all addresses (0.0.0.0)
 - * Running on http://127.0.0.1:3000
 - * Running on http://187.190.134.28:3000
 - Press CTRL+C to quit
 - * Restarting with stat

Go to Line/Column





Vamos a crear una función que convertir los datos aleatorios, es decir una lista

fakerservicio.py - proyecto_microservicio - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- PROYECTO_MICROSERVICIO
 - api
 - database
 - faker
 - proyectoweb
 - fakerservicio.py
 - web

```
1 from flask import Flask, jsonify
2 from faker import Faker
3 import itertools
4 app = Flask(__name__)
5 faker = Faker()
6 def random_data():
7     profiles = [dict(itertools.islice(faker.profile().items(), 6)) for data in range(13)]
8     return profiles
9 @app.route('/crear_registro')
10 def create_profiles():
11     try:
12         data_profiles = random_data()
13         return jsonify(data_profiles)
14         print(data_profiles)
15         return {"message": "Creando Registro"}
16     except:
17         return {"message": "Error en crear registro"}
18 if __name__ == '__main__':
19     app.run(host='0.0.0.0', port=3000, debug=True)
```

```
lg@lg-virtual-machine:~/proyecto_microservicio/faker$ source proyectoweb/bin/activate
(proyectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ ls
fakerservicio.py  proyectoweb
(proyectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$ more fakerservicio.py
from flask import Flask,jsonify
from faker import Faker
import itertools
app = Flask(__name__)
faker = Faker()
def random_data():
    profiles = [dict(itertools.islice(faker.profile().items(),6)) for data in range(13)]
    return profiles
@app.route('/crear_registro')
def create_profiles():
    try:
        data_profiles = random_data()
        return jsonify(data_profiles)
        print(data_profiles)
        return {"message":"Creando Registro"}
    except:
        return {"message":"Error en crear registro"}
if __name__=='__main__':
    app.run(host='0.0.0.0', port=3000, debug=True)
(proyectoweb) lg@lg-virtual-machine:~/proyecto_microservicio/faker$
```

about:sessionrestore

127.0.0.1:3000/crear%20reg

+

← → ↻

127.0.0.1:3000/crear registro

JSON

Raw Data

Headers

Save

Copy

Collapse All

Expand All

Filter JSON

0:

blood_group:

"AB- "

company:

"Humphrey Group"

▼ current_location:

0:

"42.1168255"

1:

"-29.216788"

job:

"Teacher, early years/pre"

residence:

"USNV Johnson\nFPO AE 49997"

ssn:

"431-46-2768"

1:

blood_group:

"0- "

company:

"Matthews Ltd"

▼ current_location:

0:

"-24.707169"

1:

"-31.455615"

job:

"Metallurgist"

residence:

"3710 Morris Park\nSouth Davidstad, AZ 04691"

ssn:

"201-23-9153"

2:

blood_group:

"0- "

company:

"King and Sons"

▼ current_location:

		127.0.0.1:3000/crear registro
JSON		Raw Data Headers
Save	Copy	Collapse All Expand All Filter JSON
0:		"71.883831"
1:		"-66.252051"
job:		"Engineer, petroleum"
residence:		"36262 Terrance Courts\nMcdowellborough, PR 19383"
ssn:		"832-18-6726"
11:		
blood_group:		"B+"
company:		"Roach-Lewis"
current_location:		
0:		"77.5340405"
1:		"24.963011"
job:		"Hydrographic surveyor"
residence:		"8465 Rodriguez Plains Suite 031\nDawnborough, WV 34553"
ssn:		"711-35-4684"
12:		
blood_group:		"A-"
company:		"Brown-Miller"
current_location:		
0:		"60.122559"
1:		"-7.968441"
job:		"Product/process development scientist"
residence:		"72695 Fitzpatrick Tunnel\nWest Sallyton, IL 72922"
ssn:		"052-74-7749"