Cómo diseñar APIs en instantes usando Flask

Sara Iris García

APIs

Las APIs (Application Programming Interface) son utilizadas para interactuar con otro software, proporcionando un conjunto de funciones y procedimientos a terceros sin exponer el código fuente.

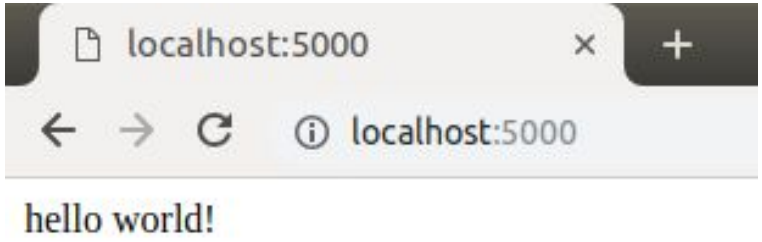
Una Web API es un tipo especial de API que utiliza el protocolo HTTP como transporte. API REST es la forma más popular de Web API.

Flask

Microframework escrito en Python para el desarrollo de aplicaciones web de forma rápida y ágil.

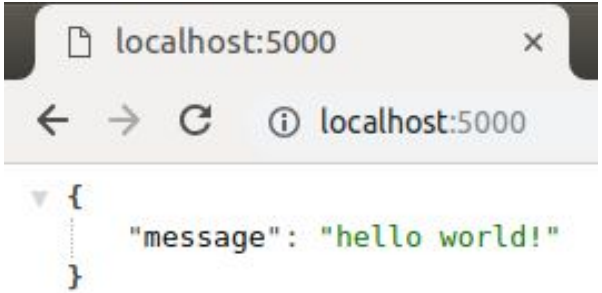
Flask

```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "hello world"

if __name__ == "__main__":
    app.run(debug=True)
```



Json response

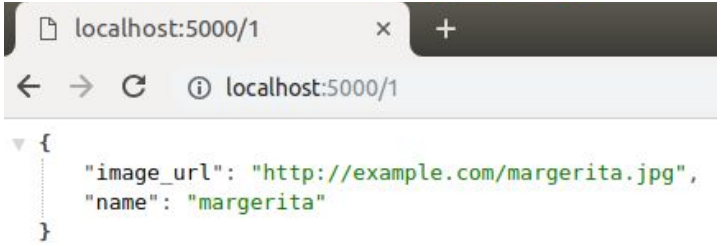
```python
from flask import Flask, jsonify
app = Flask(__name__)


@app.route("/")

def hello():
    return jsonify({"message" : "hello world!"})

if __name__ == "__main__":
    app.run(debug=True)
```



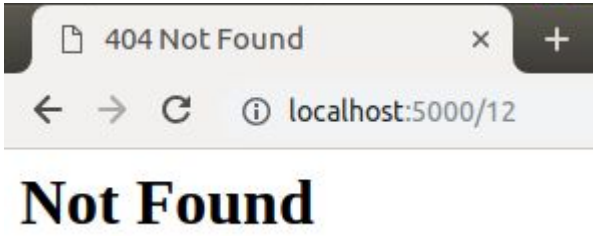
Let’s build a Pizza API!

```
from flask import Flask, jsonify
app = Flask(__name__)


@app.route("/")

def get_pizza():
    pizza = {
        "name" : "hawaian",
        "image_url" : "http://example.com/hawaian_pizza.jpg"
    }
    return jsonify(pizza)

if __name__ == "__main__":
    app.run(debug=True)
```

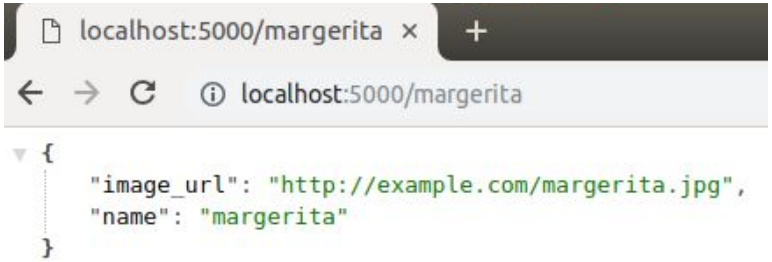
```python
from flask import Flask, jsonify
app = Flask(__name__)

pizza = [
    {
        "name" : "hawaian",
        "image_url" : "http://example.com/hawaian_pizza.jpg"
    },
    {
        "name" : "margerita",
        "image_url" : "http://example.com/margerita.jpg"
    }
]

@app.route("/<int:index>")

def get_pizza(index):

    return jsonify(pizza[index])

if __name__ == "__main__":
    app.run(debug=True)
```



```python
from flask import Flask, jsonify, abort
app = Flask(__name__)

pizza = [
    {
        "name" : "hawaian",
        "image_url" : "http://example.com/hawaian_pizza.jpg"
    },
    {
        "name" : "margerita",
        "image_url" : "http://example.com/margerita.jpg"
    }
]

@app.route("/<int:index>")

def get_pizza(index):
    try:
        return jsonify(pizza[index])
    except IndexError:
        abort(404)

if __name__ == "__main__":
    app.run(debug=True)
```



```python
from flask import Flask, jsonify, abort
app = Flask(__name__)

pizza = {
    "hawaian" : {
        "name" : "hawaian",
        "image_url" : "http://example.com/hawaian_pizza.jpg"
    },
    "margerita" : {
        "name" : "margerita",
        "image_url" : "http://example.com/margerita.jpg"
    }
}

@app.route("/<slug>")

def get_pizza(slug):
    try:
        return jsonify(pizza[slug])
    except KeyError:
        abort(404)

if __name__ == "__main__":
    app.run(debug=True)
```



Fetch from database

SQLAlchemy

```
from flask_sqlalchemy import flask_sqlalchemy

db = SQLAlchemy()

class pizza(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    slug = db.Column(db.String(30), index=True)
    name = db.Column(db.String(30), nullable=False)
    image_url = db.Column(db.String(150), nullable=False)
```

Get pizza from db

```
from flask import Flask, jsonify
from models import db, Pizza
import sys

app = Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///pizza.db"
db.init_app(app)

@app.route("/<slug>")
def get_pizza(slug):
    pizza = Pizza.query.filter(Pizza.slug==slug).first_or_404()
    my_pizza = {
        "name" : pizza.name,
        "image_url" : pizza.image_url
    }
    return jsonify(my_pizza)
```

Create and initialize db

```
if __name__ == "__main__":
    if "createdb" in sys.argv:
        with app.app_context():
            db.create_all()
        print("Database created")
    elif "seeddb" in sys.argv:
        with app.app_context():
            pizza1 = Pizza(slug="hawaian", name="hawaian", image_url="http://example.com/hawaian.jpg")
            db.session.add(pizza1)
            pizza2 = Pizza(slug="margerita", name="margerita", image_url="http://example.com/margerita.jpg")

            db.session.add(pizza2)
            db.session.commit()
        print("saved to database")
    else:
        app.run(debug=True)
```

```python
@app.route("/", methods=["POST"])
def create_pizza():
    #validate
    name = request.form.get("name")
    if not name:
        return "parameter required", 400
    image_url = request.form.get("image_url")
    if not image_url:
        return "parameter required", 400
        slug = slugify(name)

    #save in db
    pizza = Pizza(slulg=slug, name=name, image_url=image_url)
    db.session.add(pizza)
    db.session.commit(pizza)

    #response
    res = jsonify({"message" : "pizza sucessfully created"})
    res.status_code = 201
    location = url_for("get_pizza", slug=slug)
    resp.headers["Location"] = location
    return res
```

Gracias!

@sara_codes

sara.garcia@gradient.gt

@montjoile

sara-codes.com