Exemplo: Eco com Argumentos


```
from flask import Flask, abort
from flask_restful import Resource,
Api, reqparse, fields, marshal

app = Flask(__name__)
api = Api(app)

parser = reqparse.RequestParser()

parser.add_argument('msg',
    type=str,
    required=True,
    location="json",
    help="Faltou mensagem!")
```

```
class Eco:
   # como anterior...

class EcoApi(Resource):
    def get(self):
        return {'cont': eco.cont}

    def patch(self):
        args = parser.parse_args()
        return {'resp':
             eco.diga(args['msg'])}

api.add_resource(EcoApi, "/eco")

if __name__ == '__main__':
    app.run(port=1512, debug=True)
```

Prática 2: Conta Simples com ARGS

- Construa um serviço de **conta bancária simples:**
  - uma única conta (com identificador, por exemplo, “1234-5”)
- API:
  - **GET** → obtém **saldo** da conta →
    - *http://localhost:XXXX/contas/1234-5*
  - **PATCH** → **depósito** / **saque** de valores na conta →
    - *http://localhost:XXXX/contas/1234-5*
      - operação e valor no corpo da mensagem PATCH
      - (substitua XXXX pela porta do seu serviço)

Resposta (JSON):
{ "saldo": 100.0 }

Envie o seu programa Python no AVA: **Conta Simples (com param.)**.

Referências

- Roger L. Costello, "**Building Web Services the REST Way**", xFront.
- WikiPedia, "**Representational State Transfer**", https://en.wikipedia.org/wiki/Representational_state_transfer
- M. Kalin, "**Java Web Services: Up and Running**", 1st edition, O'Reilly, Feb. 2009.
- M. Elkstien, "**Learn REST: A Tutorial**", http://rest.elkstein.org/2008/02/what-is-rest.html
- M. Grimberg, "**Designing a RESTful API with Python and Flask**", http://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask