

# EP6

Carlos Josué Arciniega Noriega

05 de octubre de 2021

## Contents

<b>1</b>	<b>Introduction y Desarrollo</b>	<b>2</b>
<b>2</b>	<b>Ejemplo de funcionamiento</b>	<b>3</b>
<b>3</b>	<b>Código de la implementación</b>	<b>4</b>

# 1 Introduction y Desarrollo

Se pide un servicio web que permita realizar las 4 operaciones basicas, suponemos que son suma, resta, multiplicación y división, para lo cual se implementaron las siguientes 2 opciones:

1. Un endpoint por operación: Se tienen los endpoint "Sum"/"Res"/"Mul"/"Div"  
Los 4 reciben un json con la siguiente estructura:

$$\{'a':[n1], 'b':[n2]\}$$

Donde n1 y n2 son los numeros a operar

La respuesta por parte del servicio es la siguiente:

$$\{'r': 10.5\}$$

Donde r es el resultado de la operación.

2. Un endpoint general con la siguiente estructura:

$$"a":4.5, "b":6, "op":["op"]$$

Donde a y b son los numeros a operar y op puede tener los siguientes valores: "sum"/"res"/"mul"/"div" para indicar la operación a realizar.

La estructura de la operacion es a op b ejemplo: a/b

La respuesta por parte del servicio es la siguiente:

$$\{'r': 10.5, "status": 1\}$$

Donde r es el resultado y status puede tener los siguientes valores:

- 0:Indica que hay un error en el parámetro op
- 1:Indica que todo estuvo correcto.

## 2 Ejemplo de funcionamiento

### (a) Caso implementación 1

```
(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ python ws.py
[05/Oct/2021:13:20:55] ENGINE Listening for SIGTERM.
[05/Oct/2021:13:20:55] ENGINE Listening for SIGHUP.
[05/Oct/2021:13:20:55] ENGINE Listening for SIGUSR1.
[05/Oct/2021:13:20:55] ENGINE Bus STARTING
CherryPy Checker:
The Application mounted at '' has an empty config.

[05/Oct/2021:13:20:55] ENGINE Started monitor thread 'Autoreloader'.
[05/Oct/2021:13:20:55] ENGINE Serving on http://0.0.0.0:8080
[05/Oct/2021:13:20:55] ENGINE Bus STARTED
127.0.0.1 - - [05/Oct/2021:13:21:15] "POST /sum HTTP/1.1" 200 13 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:21:20] "POST /res HTTP/1.1" 200 12 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:21:24] "POST /mul HTTP/1.1" 200 13 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:21:28] "POST /div HTTP/1.1" 200 14 "" "curl/7.74.0"
```

```
(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":10,"b":5}' -H "Content-Type: application/json" -X POST http://localhost:8080/sum
{"r": 15}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":10,"b":5}' -H "Content-Type: application/json" -X POST http://localhost:8080/res
{"r": 5}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":10,"b":5}' -H "Content-Type: application/json" -X POST http://localhost:8080/mul
{"r": 50}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":10,"b":5}' -H "Content-Type: application/json" -X POST http://localhost:8080/div
{"r": 2.0}
```

Figure 1: Muestra de uso de endpoints

### (b) Caso implementación 2

```
(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ python ws.py
[05/Oct/2021:13:28:53] ENGINE Listening for SIGTERM.
[05/Oct/2021:13:28:53] ENGINE Listening for SIGHUP.
[05/Oct/2021:13:28:53] ENGINE Listening for SIGUSR1.
[05/Oct/2021:13:28:53] ENGINE Bus STARTING
CherryPy Checker:
The Application mounted at '' has an empty config.

[05/Oct/2021:13:28:53] ENGINE Started monitor thread 'Autoreloader'.
[05/Oct/2021:13:28:53] ENGINE Serving on http://0.0.0.0:8080
[05/Oct/2021:13:28:53] ENGINE Bus STARTED
127.0.0.1 - - [05/Oct/2021:13:29:01] "POST /calc HTTP/1.1" 200 30 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:29:07] "POST /calc HTTP/1.1" 200 30 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:29:13] "POST /calc HTTP/1.1" 200 30 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:29:19] "POST /calc HTTP/1.1" 200 30 "" "curl/7.74.0"
127.0.0.1 - - [05/Oct/2021:13:29:26] "POST /calc HTTP/1.1" 200 44 "" "curl/7.74.0"
```

```
(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":4.5,"b":0,"op":"sum"}' -H "Content-Type: application/json" -X POST http://localhost:8080/calc
{"r": 10.5, "status": 1}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":4.5,"b":0,"op":"res"}' -H "Content-Type: application/json" -X POST http://localhost:8080/calc
{"r": 27.0, "status": 1}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":4.5,"b":0,"op":"mul"}' -H "Content-Type: application/json" -X POST http://localhost:8080/calc
{"r": 0.75, "status": 1}

(base) [jos@k1] ~/SEM-2/APLICACIONES DISTRIBUIDAS/EP6/EP6
--$ curl -d '{"a":4.5,"b":0,"op":"div"}' -H "Content-Type: application/json" -X POST http://localhost:8080/calc
{"r": "Checa tu op :V", "status": 0}
```

Figure 2: Muestra de uso de endpoint

### 3 Código de la implementación

El código se puede encontrar en: [Enlace a git](#)

**Clase con métodos** myprocessor.py

```
class MyProcessor:
    def suma(self, a,b):
        return a+b
    def resta(self, a,b):
        return a-b
    def multiplica(self, a,b):
        return a*b
    def divide(self, a,b):
        return a/b
```

**Servicios** ws.py

```
import cherrypy
import pandas as pd
import myprocessor
import json
p = myprocessor.MyProcessor()

class MyWebService(object):

    @cherrypy.expose
    @cherrypy.tools.json_out()
    @cherrypy.tools.json_in()
    def sum(self):
        data = cherrypy.request.json
        out = p.suma(data['a'],data['b'])
        return json.dumps({'r':out})

    @cherrypy.expose
    @cherrypy.tools.json_out()
    @cherrypy.tools.json_in()
    def res(self):
        data = cherrypy.request.json
        out = p.resta(data['a'],data['b'])
        return json.dumps({'r':out})

    @cherrypy.expose
    @cherrypy.tools.json_out()
    @cherrypy.tools.json_in()
    def mul(self):
        data = cherrypy.request.json
        out = p.multiplica(data['a'],data['b'])
        return json.dumps({'r':out})

    @cherrypy.expose
```

```

@cherrypy.tools.json_out()
@cherrypy.tools.json_in()
def div(self):
    data = cherrypy.request.json
    out = p.divide(data['a'], data['b'])
    return json.dumps({'r': out})
@cherrypy.expose
@cherrypy.tools.json_out()
@cherrypy.tools.json_in()
def calc(self):
    data = cherrypy.request.json
    r=0
    if (data['op']=='sum'):
        r=p.suma(data['a'], data['b'])
    elif (data['op']=='res'):
        r=p.resta(data['a'], data['b'])
    elif (data['op']=='mul'):
        r=p.multiplica(data['a'], data['b'])
    elif (data['op']=='div'):
        r=p.divide(data['a'], data['b'])
    else:
        return json.dumps({'r': 'Checa tu op.:V', 'status':0})
    return json.dumps({'r':r, 'status':1})

if __name__ == '__main__':
    config = {'server.socket_host': '0.0.0.0'}
    cherrypy.config.update(config)
    cherrypy.quickstart(MyWebService())

```