

Tema 6

Remote Python Call (RPyC)



Sistemas Distribuidos
Grado en Ingeniería Informática
Universidad Carlos III de Madrid

RPyC

- Modelo de RPC para Python
- Los clientes y servidores pueden ejecutar en arquitecturas diferentes

Instalación

- Instalación en el sistema:
 - `pip install rpyc`
- Instalación para un usuario:
 - `pip install rpyc --user`

Classic RPyC

- Ejecución de un servidor clásico que puede ejecutar funciones y evaluar expresiones en Python:
 - `/usr/local/bin/rpyc_classic.py`
- La ejecución imprime por pantalla:
INFO:SLAVE/18812:server started on [127.0.0.1]:18812

Ejecución de un cliente con eval y execute (usando el servidor clásico)

client1.py

```
import rpyc
conn = rpyc.classic.connect("localhost")

conn.execute('import math')
x = conn.eval('2*math.pi')
print(x)
```

Ejecución con teleport con el servidor clásico

- Permite enviar una función al servidor para su ejecución

client2.py

```
import rpyc
conn = rpyc.classic.connect("localhost")

def square(x):
    return x**2

fn = conn.teleport(square)
print(fn(2))
```

Servicio calculadora

calc_service.py

```
import rpyc
from rpyc.utils.server import ThreadedServer
class CalculatorService(rpyc.Service):
    def exposed_add(self, a, b):
        return a + b
    def exposed_sub(self, a, b):
        return a - b
    def exposed_mul(self, a, b):
        return a * b
    def exposed_div(self, a, b):
        return a / b

if __name__ == "__main__":
    server = ThreadedServer(CalculatorService, port = 12345)
    server.start()
```

Cliente del servicio calculadora

`calc_client.py`

```
import rpyc

conn = rpyc.connect("localhost", 12345)

x = conn.root.add(4, 7)
print(x)

x = conn.root.sub(4, 7)
print(x)
```