



EBook Gratis

APRENDIZAJE zeromq

Free unaffiliated eBook created from
Stack Overflow contributors.

#zeromq

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con zeromq.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
ZeroMQ Hello World ejemplo.....	2
Creditos.....	5

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [zeromq](#)

It is an unofficial and free zeromq ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official zeromq.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con zeromq

Observaciones

Esta sección proporciona una descripción general de qué es zeromq y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de zeromq, y vincular a los temas relacionados. Dado que la Documentación para zeromq es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Zeromq tiene una gran cantidad de enlaces para diferentes idiomas.

Para encontrar el enlace correcto y las instrucciones de uso, [siga esto](#) .

También puedes usar: una [implementación C #](#) pura o una [implementación Java](#)

ZeroMQ Hello World ejemplo.

En este ejemplo, crearemos un cliente y un servidor sencillos con sockets REQ-REP (solicitud-respuesta). El cliente envía "Hola" al servidor, que responde con "Mundo". El servidor abre un ZeroMQ REP-socket en el puerto 5555, lee las solicitudes en él y responde con "Mundo" a cada solicitud.

Hola servidor mundial en C:

```
#include <zmq.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <assert.h>

int main (void)
{
    // Socket to talk to clients
    void *context = zmq_ctx_new ();
    void *responder = zmq_socket (context, ZMQ_REP);
    int rc = zmq_bind (responder, "tcp://*:5555");
    assert (rc == 0);

    while (1) {
        char buffer [10];
        zmq_recv (responder, buffer, 10, 0);
        printf ("Received Hello\n");
        sleep (1); // Do some 'work'
        zmq_send (responder, "World", 5, 0);
    }
}
```

```
    }  
    return 0;  
}
```

Hola servidor mundial en Java:

```
import org.zeromq.ZMQ;  
  
public class hwserver {  
  
    public static void main(String[] args) throws Exception {  
        ZMQ.Context context = ZMQ.context(1);  
  
        // Socket to talk to clients  
        ZMQ.Socket responder = context.socket(ZMQ.REP);  
        responder.bind("tcp://*:5555");  
  
        while (!Thread.currentThread().isInterrupted()) {  
            // Wait for next request from the client  
            byte[] request = responder.recv(0);  
            System.out.println("Received Hello");  
  
            // Do some 'work'  
            Thread.sleep(1000);  
  
            // Send reply back to client  
            String reply = "World";  
            responder.send(reply.getBytes(), 0);  
        }  
        responder.close();  
        context.term();  
    }  
}
```

Hola cliente mundial en C:

```
#include <zmq.h>  
#include <string.h>  
#include <stdio.h>  
#include <unistd.h>  
  
int main (void)  
{  
    printf ("Connecting to hello world server...\n");  
    void *context = zmq_ctx_new ();  
    void *requester = zmq_socket (context, ZMQ_REQ);  
    zmq_connect (requester, "tcp://localhost:5555");  
  
    int request_nbr;  
    for (request_nbr = 0; request_nbr != 10; request_nbr++) {  
        char buffer [10];  
        printf ("Sending Hello %d...\n", request_nbr);  
        zmq_send (requester, "Hello", 5, 0);  
        zmq_recv (requester, buffer, 10, 0);  
        printf ("Received World %d\n", request_nbr);  
    }  
    zmq_close (requester);  
    zmq_ctx_destroy (context);  
    return 0;  
}
```

```
}
```

Hola cliente mundial en Java:

```
import org.zeromq.ZMQ;

public class hwclient {

    public static void main(String[] args) {
        ZMQ.Context context = ZMQ.context(1);

        // Socket to talk to server
        System.out.println("Connecting to hello world server...");

        ZMQ.Socket requester = context.socket(ZMQ.REQ);
        requester.connect("tcp://localhost:5555");

        for (int requestNbr = 0; requestNbr != 10; requestNbr++) {
            String request = "Hello";
            System.out.println("Sending Hello " + requestNbr);
            requester.send(request.getBytes(), 0);

            byte[] reply = requester.recv(0);
            System.out.println("Received " + new String(reply) + " " + requestNbr);
        }
        requester.close();
        context.term();
    }
}
```

Nota: los sockets REQ-REP zeromq están bloqueando, por lo que cuando un cliente (REQ) envía un mensaje, no puede hacer nada más hasta que reciba una respuesta del servidor y viceversa, hasta que el servidor (REP) reciba un mensaje de El cliente, no puede enviarle nada.

Lea Empezando con zeromq en línea: <https://riptutorial.com/es/zeromq/topic/9431/empezando-con-zeromq>

Creditos

S. No	Capítulos	Contributors
1	Empezando con zeromq	Community , DontPanic