



**EBook Gratuito**

# APPENDIMENTO

## zeromq

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#zeromq**

# Sommario

|  |          |
|--|----------|
| Di.....                                      | 1        |
| <b>Capitolo 1: Iniziare con zeromq</b> ..... | <b>2</b> |
| Osservazioni.....                            | 2        |
| Examples.....                                | 2        |
| Installazione o configurazione.....          | 2        |
| Esempio ZeroMQ Hello World.....              | 2        |
| <b>Titoli di coda</b> .....                  | <b>5</b> |

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](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con zeromq

## Osservazioni

Questa sezione fornisce una panoramica di cosa sia Zeromq e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di zeromq e collegarsi agli argomenti correlati. Poiché la documentazione di zeromq è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

## Examples

### Installazione o configurazione

Zeromq ha un numero enorme di associazioni per diverse lingue.

Per trovare il legame giusto e le istruzioni su come usare [segui questo](#).

Inoltre è possibile utilizzare: [un'implementazione C #](#) pura o [un'implementazione Java](#)

### Esempio ZeroMQ Hello World.

In questo esempio creeremo un semplice client e server con socket REQ-REP (richiesta-risposta). Il client invia "Hello" al server, che risponde con "World". Il server apre un socket REP ZeroMQ sulla porta 5555, legge richieste su di esso e risponde con "World" a ciascuna richiesta.

#### Ciao server World in 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;
}
```

## Ciao server World in 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();
    }
}
```

## Ciao cliente del mondo in 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;
}
```

## Ciao cliente del mondo in 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:** i socket zeromq REQ-REP stanno bloccando, quindi quando un client (REQ) invia un messaggio, non può fare nient'altro finché non riceve una risposta dal server e viceversa - finché il server (REP) non riceve un messaggio da il cliente, non può mandargli nulla.

Leggi Iniziare con zeromq online: <https://riptutorial.com/it/zeromq/topic/9431/iniziare-con-zeromq>

---

# Titoli di coda

| S. No | Capitoli            | Contributors  |
|-------|---------------------|---|
| 1     | Iniziare con zeromq | <a href="#">Community</a> , <a href="#">DontPanic</a> |