



**FREE eBook**

# LEARNING

---

## zeromq

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

**#zeromq**

# Table of Contents

About.....	1
Chapter 1: Getting started with zeromq.....	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
ZeroMQ Hello World example.....	2
Credits.....	5

---

# About

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)

---

# Chapter 1: Getting started with zeromq

## Remarks

This section provides an overview of what zeromq is, and why a developer might want to use it.

It should also mention any large subjects within zeromq, and link out to the related topics. Since the Documentation for zeromq is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Zeromq has a huge number of bindings for different languages.

To find the right binding and instructions how to use [follow this](#).

Also you can use: a pure [C# implementation](#) or [Java implementation](#)

### ZeroMQ Hello World example.

In this example we will make a simple client and server with REQ-REP (request-reply) sockets. The client sends "Hello" to the server, which replies with "World". Server opens a ZeroMQ REP-socket on port 5555, reads requests on it, and replies with "World" to each request.

### Hello World server 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;
}
```

## Hello World server 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();
    }
}
```

## Hello World client 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;
}
```

## Hello World client 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();
    }
}

```

**Note:** REQ-REP zeromq sockets are blocking, so when a client (REQ) sends a message, he can't do anything else until he receives a response from the server and vice versa - until the server (REP) receives a message from the client, he can't send him anything.

Read [Getting started with zeromq online](https://riptutorial.com/zeromq/topic/9431/getting-started-with-zeromq): <https://riptutorial.com/zeromq/topic/9431/getting-started-with-zeromq>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with zeromq	<a href="#">Community</a> , <a href="#">DontPanic</a>