Router

while (count++<50) {

char identity[100] = { 0 };

char id[100] = { 0 };

int ret = zmq\_recv(frontend, identity, 100, 0);

if (ret != -1) {

printf("identity ==== %s\n", identity);

strcpy(id, identity);

}

memset(identity, 0, 100);

ret = zmq\_recv(frontend, identity, 100, 0);

if (ret != -1) {

printf("empty ==== %s\n", identity);

}

zmq\_msg\_t msg\_recv;

zmq\_msg\_init(&msg\_recv);

ret = zmq\_msg\_recv(&msg\_recv, frontend, 0);

if (ret != -1) {

printf("content ==== %s\n", (char \*)zmq\_msg\_data(&msg\_recv));

}

Sleep(1000);

char send[40] = { 0 };

sprintf(send, "%s%s", id, "main hello");

zmq\_send(frontend, id, strlen(id), ZMQ\_SNDMORE);

zmq\_send(frontend, "", 0, ZMQ\_SNDMORE);

zmq\_send(frontend, "main hello", 6, 0);

}

while (count<100) {

int centitick;

for (centitick = 0; centitick < 100; centitick++) {

zmq\_poll(items, 1, 1);

if (items[0].revents & ZMQ\_POLLIN)

{

zmq\_msg\_t msg\_recv;

zmq\_msg\_init(&msg\_recv);

int ret = zmq\_msg\_recv(&msg\_recv, client, 0);

if (ret != -1) {

printf("empty ==== %s\n", (char \*)zmq\_msg\_data(&msg\_recv));

}

ret = zmq\_msg\_recv(&msg\_recv, client, 0);

if (ret != -1) {

printf("router content ==== %s\n", (char \*)zmq\_msg\_data(&msg\_recv));

}

zmq\_msg\_close(&msg\_recv);

}

}

count++;

zmq\_msg\_init(&msg\_send);

zmq\_send(client, "", 0, ZMQ\_SNDMORE);

if (reader.parse(in, value))

{

printf("serip ==== %s \n", value["msgHead"]["serverIP"].toStyledString().data());

}

int len = strlen(value.toStyledString().data());

char \*strsend = (char \*)malloc(len + 1);

memset(strsend, 0, len + 1);

strcpy(strsend, value.toStyledString().data());

char str[20] = { 0 };

sprintf(str, "request #%d \n", ++request\_nbr);

char \*send = (char \*)malloc(len + 1 + 21);

memset(send, 0, len + 1 + 21);

strcat(send, str);

strcat(send, strsend);

strcpy(strsend, "dealer content");

memcpy(zmq\_msg\_data(&msg\_send), str, strlen(str));

printf("strsend len ============ %d \n", strlen(strsend));

zmq\_send(client, strsend, strlen(strsend) + 1, 0);

free(strsend);

free(send);

////

// Sleep(5000);

}