

TencentDB for Redis

Tutorials

Product Documentation



Copyright Notice

©2013–2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Tutorials

Access Connection

Tutorials

Access Connection

Last updated : 2017-06-28 17:27:58

Take Clink codes as an example

Preparation before running: Download and install hiredis at <https://github.com/redis/hiredis>

Sample Codes:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>

int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;

    if (argc < 4) {
        printf("Usage: 192.168.0.195 6379 instance_id password\n");
        exit(0);
    }
    const char *hostname = argv[1];
    const int port = atoi(argv[2]);
    const char *instance_id = argv[3];
    const char *password = argv[4];

    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    c = redisConnectWithTimeout(hostname, port, timeout);
    if (c == NULL || c->err) {
        if (c) {
            printf("Connection error: %s\n", c->errstr);
            redisFree(c);
        } else {
            printf("Connection error: can't allocate redis context\n");
        }
        exit(1);
    }

    /* AUTH */
    reply = redisCommand(c, "AUTH %s:%s", instance_id, password);
    printf("AUTH: %s\n", reply->str);
    freeReplyObject(reply);
}
```

```
/* PING server */
reply = redisCommand(c, "PING");
printf("PING: %s\n", reply->str);
freeReplyObject(reply);

/* Set a key */
reply = redisCommand(c, "SET %s %s", "name", "credis_test");
printf("SET: %s\n", reply->str);
freeReplyObject(reply);

/* Try a GET */
reply = redisCommand(c, "GET name");
printf("GET name: %s\n", reply->str);
freeReplyObject(reply);

/* Disconnects and frees the context */
redisFree(c);

return 0;
}
```