

云数据库 KeeWiDB

SDK 参考



腾讯云

【版权声明】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100 或 95716。

文档目录

[SDK 参考](#)

[PHP 连接示例](#)

[Java 连接示例](#)

[Node.js 连接示例](#)

[Python 连接示例](#)

[C 连接示例](#)

[Go 连接示例](#)

[.Net 连接示例](#)

SDK 参考

PHP 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 KeeWiDB 控制台 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 运行前必备：下载客户端 [phpredis](#)。

示例代码

```
<?php
/**以下参数分别填写您的 KeeWiDB 实例内网 IP、端口号、实例 ID 和密码*/
$host = "192.xx.xx.2";
$port = 6379;
$pwd = "123tj6na";

$redis = new Redis();
//连接 Redis
if ($redis->connect($host, $port) == false) {
    die($redis->getLastError());
}

//鉴权
if ($redis->auth($pwd) == false) {
    die($redis->getLastError());
}

/**接下来可以开始操作 KeeWiDB 实例，可以参考
https://github.com/phpredis/phpredis */

//设置 Key
if ($redis->set("redis", "tencent") == false) {
    die($redis->getLastError());
}
```

```
echo "set key redis suc, value is:tencent\n";  
  
//获取 Key  
$value = $redis->get("redis");  
echo "get key redis is:".$value."\n";  
?>
```

运行结果

```
[root@VM_0_3_centos bin]# ./php Test.php  
set key redis suc, value is:tencent  
get key redis is:tencent
```

Java 连接示例

最近更新时间：2024-10-18 15:10:22

本文列举客户端 Java 代码示例，辅助您使用 SSL 加密或不加密的方式访问数据库。

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 [KeeWiDB 控制台](#) 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4 地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 下载 Jar 包 [Jedis](#)，及其相关依赖（[gson](#), [slf4j](#)）。建议使用最新版本。

连接示例

您需要根据注释修改参数：连接数据库的 IP、端口及账号密码信息。

```
import redis.clients.jedis.Jedis;

public class HelloRedis {

    public static void main(String[] args) {
        try {
            /**以下参数，分别填写您的 KeeWiDB 实例内网 IP、端口号、实例 ID 和密码。*/
            String host = "192.xx.xx.195";
            int port = 6379;
            String instanceid = "crs-09xxxqv";
            String password = "123ad6aq";
            //连接 Redis
            Jedis jedis = new Jedis(host, port);
            //鉴权
            jedis.auth(instanceid + ":" + password);

            /**操作 KeeWiDB 实例，可以参考
            https://github.com/xetorthio/jedis */
            //设置 Key
            jedis.set("redis", "tencent");
            System.out.println("set key redis suc, value is: tencent");
        }
    }
}
```

```
//获取 Key
String value = jedis.get("redis");
System.out.println("get key redis is: " + value);

//关闭退出
jedis.quit();
jedis.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

}
```

运行结果

```
[root@VM_0_194_centos bin]# ./java -cp jedis-2.4.2.jar:. HelloRedis
set key redis suc, value is: tencent
get key redis is: tencent
[root@VM_0_194_centos bin]#
```

Node.js 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 KeeWiDB 控制台 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 安装 node-redis。
 - V4 之前版本：执行 `npm install redis@3.1.2 --save` 命令安装 node-redis。
 - V4 版本：执行 `npm install redis --save` 命令安装 node-redis。

示例代码

V4版本之前的代码示例：

```
var redis = require("redis");

/**以下参数分别填写您的 KeeWiDB 实例内网 IP、端口号、实例 ID 和密码*/
var host = "192.xx.xx.2",
port = "6379",
instanceid = "c53xx52f-55dc-4c22-a941-630xxx88",
pwd = "12as6zb";
//连接
var client = redis.createClient(port, host, {detect_buffers: true});
//连接错误
client.on("error", function(error) {
    console.log(error);
});
//鉴权
client.auth(instanceid + ":" + pwd);

/**接下来可以开始操作 KeeWiDB 实例 */
//设置 Key
client.set("redis", "tencent", function(err, reply){
    if (err) {
        console.log(err);
    }
});
```

```
        return;
    }

    console.log("set key redis " + reply.toString() + ", value is
tencent");
});

//获取 Key
client.get("redis", function (err, reply) {
    if (err) {
        console.log(err);
        return;
    }
    console.log("get key redis is:" + reply.toString());
//程序结束关闭客户端
    client.end();
});
}
```

V4版本代码示例：

```
const redis = require('redis')

/**以下参数分别填写您的 KeeWiDB 实例内网 IP、端口号、实例 ID 和密码*/
var host = "192.xx.xx.2",
port = "6379",
instanceid = "cc53xx52f-55dc-4c22-a941-630xxx88",
pwd = "12as6zb";

;(async () =>{
    const redisClient = redis.createClient({
        socket: {
            port: port,
            host: host
        },
        password: instanceid + ":" + pwd
    });

    redisClient.on('ready', () => {
        console.log('redis is ready...')
    })

    redisClient.on('error', err => {
        console.log(err)
    })
})
```

```
}

await redisClient.connect() // 连接

/* 增 改 */
const status = await redisClient.set('KeeWiDB', 'tencent') // 设置值
console.log(status)

/* 查 */
const value = await redisClient.get('KeeWiDB') // 得到value 没有则为
null
console.log(value)

await redisClient.quit() // 关闭
})();
```

运行结果

```
[root@VM_0_3_centos bin]# ./node Test.js
set key redis suc, value is:OK
get key redis is:tencent
```

Python 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 [KeeWiDB 控制台](#) 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4 地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 下载并安装 [redis-py](#), 推荐使用最新版本。

连接示例

您需要根据注释修改参数：连接数据库的 IP、端口及账号密码信息。

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-
import redis

#这里替换为连接的实例 host 和 port
host = '192.xx.xx.195'
port = 6379

#这里替换为实例 ID 和实例 password
user='username'
pwd='password'

#连接时通过 password 参数指定 AUTH 信息，默认账号连接，password 为 pwd。
r = redis.StrictRedis(host=host, port=port, password=user+'@'+pwd)

#连接建立后就可以进行数据库操作，请参见 https://github.com/andymccurdy/redis-py
r.set('name', 'python_test');
print(r.get('name'))
```

运行结果

```
[root@VM_0_194_centos fasterquan]# python redis-python.py
python_test
[root@VM_0_194_centos fasterquan]#
```


C 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 KeeWiDB 控制台 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 下载并安装 [hireedis](#)。

连接示例

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

#include <hireedis/hiredis.h>

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

    if (argc < 4) {
        printf("Usage: 192.xx.xx.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) {
            free(c);
        }
        exit(1);
    }

    if (redisCommand(c, "SET %s %s", instance_id, password) != -1) {
        printf("Set instance_id to %s successfully.\n", instance_id);
    } else {
        printf("Failed to set instance_id.\n");
    }

    redisFree(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", 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;
}
```

运行结果

```
[root@VM_0_194_centos hiredis]# ./example 192 168 195 6379 84ffd722-b506-4934
-9025-645bb2a0997b 1234567q
AUTH: OK
PING: PONG
SET: OK
GET name: credis_test
[root@VM_0_194_centos hiredis]#
```

Go 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 KeeWiDB 控制台 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 下载客户端 [Go-redis](#)。

示例代码

```
package main
import (
    "fmt"
    "github.com/go-redis/redis"
    "log"
)
func main() {
    const host="192.xx.xx.195"
    const port="6379"
    const instanceId="84ffd722-b506-4934-9025-64xxx997b"
    const pass="123d7sq"
    // 连接 KeeWiDB 服务器 192.xx.xx.195:6379 并授权 instanceId 密码
    client:= redis.NewClient(&redis.Options{
        Addr: host + ":" + port,
        Password: instanceId + ":" + pass,
    })
    newvalue :=[]byte("KeeWiDB!");
    var err=client.Set("tencent",newvalue,0).Err();
    if err != nil { // 设置值出错
        log.Println(err)
        return
    }
    value, err := client.Get("tencent").Result() // 取值
    if err != nil {
```

```
    log.Println(err)
    return
}
fmt.Println("tencent value is:", fmt.Sprintf("%s", value)) //输出
}
```

运行结果

```
[root@VM_0_194_centos go_src]# go run testRedis.go
name value is: QcloudV5!
[root@VM_0_194_centos go_src]#
```

.Net 连接示例

最近更新时间：2024-10-18 15:10:22

说明：

如果业务层代码使用了 web 框架，可以参考框架访问单机版 redis 的方法来接入 Keewidb。

准备工作

- 在 KeeWiDB 控制台 的实例详情页面的网络信息区域，获取连接数据库的内网 IPv4地址及端口。具体信息，请参见 [查看实例详情](#)。
- 已获取访问数据库的账号与密码。
- 下载并安装 [ServiceStack.Redis](#)。

示例代码

不使用连接池

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string host = "10.xx.xx.46"; // 实例访问 host 地址
            int port = 6379; // 端口信息
            string instanceId = "bd87dadc-8xx1-4xx1-86dd-021xxxcd96"; // 实
            例 ID
            string pass = "1234567q"; // 密码

            RedisClient redisClient = new RedisClient(host, port,
instanceId + ":" + pass);
            string key = "name";
            string value = "QcloudV5!";
            redisClient.Set(key, value); // 设置值
        }
    }
}
```

```
        System.Console.WriteLine("set key:[" + key + "]value:[" + value + "]");
        string getValue =
System.Text.Encoding.Default.GetString(redisClient.Get(key)); //读取值
        System.Console.WriteLine("value:" + getValue);
        System.Console.Read();
    }
}
}
```

使用 ServiceStack 4.0 连接池

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] testReadWriteHosts = new[] {
"redis://:fb92bxxxabf11e5:1234xx8a1A@10.x.x.1:6379/*redis://:实例ID:密码@访问地址:端口*/
};

RedisConfig.VerifyMasterConnections = false; //需要设置
PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10/*连接池个数*/,
10/*连接池超时时间*/, testReadWriteHosts);
for (int i = 0; i < 100; i++)
{
    IRedisClient redisClient =
redisPoolManager.GetClient(); //获取连接
    RedisNativeClient redisNativeClient =
(RedisNativeClient)redisClient;
    redisNativeClient.Client = null; //需要设置
    try
    {
        string key = "test1111";
    }
}
```

```
        string value = "test1111";
        redisClient.Set(key, value);
        redisClient.Dispose(); //
    }
    catch (Exception e)
    {
        System.Console.WriteLine(e.Message);
    }
}
System.Console.Read();
}
}
```

使用 ServiceStack 3.0 连接池

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] testReadWriteHosts = new[] {
                "fb92bfxxbf11e5:123456xx1A@10.x.x.1:6379" /*实例ID:密码@访问地址:端口*/
            };
            PooledRedisClientManager redisPoolManager = new
PooledRedisClientManager(10/*连接池个数*/, 10/*连接池超时时间*/, testReadWriteHosts);
            for (int i = 0; i < 100; i++)
            {
                IRedisClient redisClient =
redisPoolManager.GetClient(); //获取连接
                try
                {
                    string key = "test1111";
                    string value = "test1111";
                }
            }
        }
    }
}
```

```
        redisClient.Set(key, value);
        redisClient.Dispose();//
    }
    catch (Exception e)
    {
        System.Console.WriteLine(e.Message);
    }
}
System.Console.Read();
}
```

运行结果

```
set key:[name]value:[QcloudV5!]
value:"QcloudV5!"
```