

# 云数据库 MongoDB

## SDK 参考

### 产品文档



腾讯云

**【版权声明】**

©2013-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【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。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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。

---

## 文档目录

### SDK 参考

Shell连接示例

PHP 连接示例

Node.js连接示例

Java 连接示例

Python连接示例

Python 读写示例

Go 连接示例

PHP 重连示例

# SDK 参考

## Shell连接示例

最近更新时间：2024-01-12 11:02:42

在云服务器 CVM 中可用 MongoDB 提供的 shell 客户端（[查看安装文档](#)）连接云数据库 MongoDB 进行数据管理，请注意使用最新版本的 MongoDB 客户端套件。

### 快速开始

典型的连接命令如下：

```
mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
```

### 注意：

使用连接串访问 MongoDB 时，如密码包含特殊字符，则特殊字符需要进行转义成 URL 编码才能正常识别，例如，@需要转义为%40。

如图：

```
#: ./mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/admin
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> show dbs
admin    0.031GB
local   3.030GB
testdb  0.031GB
mongos> use testdb
switched to db testdb
mongos> show collections
system.indexes
testcollection
mongos> db.testcollection.find().limit(2)
```

### 多种认证方式的连接说明

在 [连接实例](#) 有说明，云数据库 MongoDB 默认提供 rwuser 和 mongouser 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式。

对于这两种认证方式，shell 的参数是不一样的，具体请看下文。

### SCRAM-SHA-1 认证 (mongouser)

默认用户 `mongouser` 以及在控制台创建的所有新用户都使用 **SCRAM-SHA-1** 认证，其 shell 连接参数与 [快速开始](#) 章节完全一样，无需添加额外参数，示例如下：

```
mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
```

如果您希望连接 MongoDB 服务后直接进入到一个 db，例如 `singer`，请按示例操作：

```
mongo 10.66.187.127:27017/singer -u mongouser -p thepasswordA1 --
authenticationDatabase admin
```

如图：

```
#: ./mongo 10.66.187.127:27017/singer -u mongouser -p thepasswordA1 --authenticationDatabase ad
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/singer
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> db
singer
mongos> █
```

## MONGODB-CR 认证 (rwuser)

请注意，只有默认用户 `rwuser` 使用 **MONGODB-CR** 认证，其 shell 连接参数需要指明认证方式为 **MONGODB-CR**，示例如下：

```
mongo 10.66.187.127:27017/admin -u rwuser -p thepasswordA1 --
authenticationMechanism=MONGODB-CR
```

如图：

```
#: ./mongo 10.66.187.127:27017/admin -u rwuser -p thepasswordA1 --authenticationMechanism=MONGOD
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/admin
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> show dbs
admin 0.031GB
local 3.030GB
testdb 0.031GB
mongos> █
```

如果您希望连接 MongoDB 服务后直接进入到一个 db，例如 `singer`，请按示例操作：

```
mongo 10.66.187.127:27017/singer -u rwuser -p thepasswordA1 --
authenticationMechanism=MONGODB-CR --authenticationDatabase admin
```

如图：

```
#: ./mongo 10.66.187.127:27017/singer -u rwuser -p thepasswordA1 --authenticationMechanism=MONGODB-CR
authenticationDatabase admin
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/singer
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> db
singer
mongos>
```

## 使用 shell 进行数据导入和导出

上文所述的两种认证方式都可以在 shell 里进行数据导入和导出，请参见 [导出导入](#)。

# PHP 连接示例

最近更新时间：2025-01-06 16:02:32

## 相关说明

云数据库 MongoDB 默认提供 `rwuser` 和 `mongouser` 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，对于这两种认证方式，连接 URI 需要做不同的处理，具体参见 [连接实例](#)。

在 PHP 里，有驱动可用于连接操作 MongoDB 数据库：`mongodb` ([PHP 官网文档](#)) - MongoDB 官方推荐 `mongodb` 驱动，但需要 PHP 5.4 以上版本。

下面用上述驱动演示连接云数据库 MongoDB 并进行读写。

## 使用 `mongodb` 驱动

`mongodb` 安装方法参考 [官方安装步骤](#)。

`mongodb` 驱动可以用 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，具体参见 [连接实例](#)。

示例代码：

```
<?php
// 拼接连接 URI
$uri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin';
$manager = new MongoDB\Driver\Manager($uri);

// 准备写入数据
$document1 = [
    'username' => 'lily',
    'age'      => 34,
    'email'    => 'lily@qq.com'
];

// 驱动预处理数据，这里可以看到 MongoDB 的 _id 是驱动生成的
$bulk = new MongoDB\Driver\BulkWrite;
$_id1 = $bulk->insert($document1);

$result = $manager->executeBulkWrite('tsdb.table1', $bulk);

// 或者根据实际需要下面的代码确保数据写入到大多数节点
// $writeConcern = new
MongoDB\Driver\WriteConcern(MongoDB\Driver\WriteConcern::MAJORITY, 1000);
// $result = $manager->executeBulkWrite('testdb.testcollection', $bulk,
$writeConcern);

// 查询
$filter = ['_id' => $_id1];
$query = new MongoDB\Driver\Query($filter);
```

```
$rows = $manager->executeQuery('tsdb.table1', $query); // 也可选择优先从从库读
foreach($rows as $r){
    print_r($r);
}
```

输出：

```
stdClass Object
(
    [_id] => MongoDB\BSON\ObjectID Object
        (
            [oid] => 582c001618c90a16363abc31
        )

    [username] => lily
    [age] => 34
    [email] => lily@qq.com
)
```

## 推荐使用 PHPLIB 库（基于 mongodb 驱动封装）

使用 mongodb 驱动推荐搭配 [PHPLIB](#) 使用，[查看相关文档](#)。

PHPLIB 的安装方法参考 [官方安装步骤](#)，请注意 PHPLIB 依赖与 mongodb 驱动。

示例代码：

```
<?php
require_once __DIR__ . "/vendor/autoload.php";

// 初始化
$mongoClient = new
MongoDB\Client('mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin');

// 使用 demo 库下的 users 集合
$collection = $mongoClient->demo->users;

// 写入一条数据
$insertOneResult = $collection->insertOne(['name' => 'gomez']);

printf("Inserted %d document(s)\n", $insertOneResult->getInsertedCount());
var_dump($insertOneResult->getInsertedId());

// 查询数据
$document = $collection->findOne(['name' => 'gomez']);
```



```
var_dump($document);
```

输出：

```
Inserted 1 document(s)
object (MongoDB\\BSON\\ObjectID) #11 (1) {
    ["oid"]=>
    string(24) "57e3bf20bf605714a53e69c1"
}
object (MongoDB\\Model\\BSONDocument) #16 (1) {
    ["storage": "ArrayObject":private]=>
    array(2) {
        ["_id"]=>
        object (MongoDB\\BSON\\ObjectID) #14 (1) {
            ["oid"]=>
            string(24) "57e3bf20bf605714a53e69c1"
        }
        ["name"]=>
        string(5) "gomez"
    }
}
```

# Node.js连接示例

最近更新时间：2024-01-12 11:03:23

## 相关说明

云数据库 MongoDB 默认提供 `rwuser` 和 `mongouser` 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，对于这两种认证方式，连接 URI 需要做不同的处理，具体参见 [连接实例](#)。

[Node.js MongoDB 驱动文档](#)

## 快速开始

### Node.js 原生示例代码

Shell 安装驱动包：

```
npm install mongodb --save
( 如遇安装不成功可以尝试更换源, npm config set registry http://registry.cnpmjs.org )
npm init
```

程序代码：

```
'use strict';

var MongoClient = require('mongodb').MongoClient,
    assert = require('assert');

// 拼接 URI
var url = 'mongodb://mongouser:thepasswordA1@10.66.161.177:27017/admin';

mongoClient.connect(url, function(err, db) {
    assert.equal(null, err);
    var db = db.db('testdb'); // 选择一个 db
    var col = db.collection('demoCol'); // 选择一个集合(表)
    // 插入数据
    col.insertOne(
        {
            a: 1,
            something: "yy"
        },
        //可选参数
        //{
        //    w: 'majority' // 开启“大多数”模式，保证数据写入 Secondary 节点
        }
    );
});
```

```
    //},  
    function(err, r) {  
        console.info("err:", err);  
        assert.equal(null, err);  
        // 断言写入成功  
        assert.equal(1, r.insertedCount);  
        // 查询数据  
        col.find().toArray(function(err, docs) {  
            assert.equal(null, err);  
            console.info("docs:", docs);  
            db.close();  
        });  
    }  
);  
});
```

输出：

```
[root@VM_2_167_centos node]# node index.js  
docs: [ { _id: 567a1bf26773935b3ff0b42a, a: 1, something: 'yy' } ]
```

## Node.js mongoose 连接示例

```
var dbUri = "mongodb://" + user + ":" + password + "@" + host + ":" + port +  
"/" + dbName;  
var opts = {  
  auth: {  
    authMechanism: 'MONGODB-CR', // 如果使用 SCRAM-SHA-1 认证则不需要此参数  
    authSource: 'admin'  
  }  
};  
var connection = mongoose.createConnection(dbUri, opts);
```

# Java 连接示例

最近更新时间：2024-01-12 11:03:39

## 相关说明

云数据库 MongoDB 默认提供 `rwuser` 和 `mongouser` 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，对于这两种认证方式，连接 URI 需要做不同的处理，具体参见 [连接实例](#)。

[Java MongoDB 驱动文档](#)

[Java Jar 包下载](#)，请选择3.2以上版本

## 快速开始

### 原生 Java 示例代码

```
package mongodbdemo;

import org.bson.*;
import com.mongodb.*;
import com.mongodb.client.*;

public class Mongodbdemo {

    public static void main(String[] args) {
        String mongoUri =
"mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin"; // 多 IP 实例
Uri 可参考：
mongodb://mongouser:*****@172.xx.xx.124:27017,172.xx.xx.27:27017,172.xx.xx.136
:27017/test?authSource=admin&replicaSet=cmgo-fsstfgob_0
        MongoClientURI connStr = new MongoClientURI(mongoUri);
        MongoClient mongoClient = new MongoClient(connStr);
        try {
            // 使用名为 someonedb 的数据库
            MongoDBDatabase database = mongoClient.getDatabase("someonedb");
            // 取得集合/表 someonetable 句柄
            MongoCollection<Document> collection =
database.getCollection("someonetable");

            // 准备写入数据
            Document doc = new Document();
            doc.append("key", "value");
            doc.append("username", "jack");
```

```
doc.append("age", 31);

// 写入数据
collection.insertOne(doc);
System.out.println("insert document: " + doc);

// 读取数据
BsonDocument filter = new BsonDocument();
filter.append("username", new BsonString("jack"));
MongoCursor<Document> cursor = collection.find(filter).iterator();
while (cursor.hasNext()) {
    System.out.println("find document: " + cursor.next());
}
} finally {
    //关闭连接
    mongoClient.close();
}
}
```

输出：

```
INFO: Opened connection [connectionId{localValue:2, serverValue:67621}] to
10.66.122.28:27017
insert document: Document{{key=value, username=jack, age=31,
_id=56a6ebb565b33b771f9826dd}}
find document: Document{{_id=56a3189565b33b2e7ca150ba, key=value,
username=jack, age=31}}
Jan 26, 2016 11:44:53 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:67621}] to
10.66.122.28:27017 because the pool has been closed.
```

## Spring Data MongoDB 配置示例

本示例主要为了体现出 [认证库 admin](#) 的配置方法，具体还请参考您使用的 Spring 和 Spring Data MongoDB 的版本而定。

```
<bean id="mongoTemplate"
class="org.springframework.data.mongodb.core.MongoTemplate">
    <constructor-arg name="mongoDbFactory" ref="mongoDbFactory" />
</bean>
<bean id="mongoDbFactory"
class="org.springframework.data.mongodb.core.SimpleMongoDbFactory">
    <constructor-arg name="mongo" ref="mongo" />
    <constructor-arg name="databaseName" value="您的目标库" />
    <constructor-arg name="credentials" ref="userCredentials" />
    <constructor-arg name="authenticationDatabaseName" value="admin" />
```

```
</bean>
<bean id="userCredentials"
class="org.springframework.data.authentication.UserCredentials">
    <constructor-arg name="username" value="用户名" />
    <constructor-arg name="password" value="密码" />
</bean>
```

# Python连接示例

最近更新时间：2024-01-12 11:03:51

## 相关说明

云数据库 MongoDB 默认提供 `rwuser` 和 `mongouser` 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，对于这两种认证方式，连接 URI 需要做不同的处理，具体参见 [连接实例](#)。

[Python 驱动下载](#)。具体信息，请参见 [MongoDB Python Drivers](#)。

## 快速开始

### Python 示例代码 1

```
#!/usr/bin/python
import pymongo
import random

mongodbUri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin'

client = pymongo.MongoClient(mongodbUri)
db = client.somedb
db.user.drop()
element_num=10
for id in range(element_num):
    name = random.choice(['R9', 'cat', 'owen', 'lee', 'J'])
    sex = random.choice(['male', 'female'])
    db.user.insert_one({'id':id, 'name':name, 'sex':sex})

content = db.user.find()
for i in content:
    print i
```

### Python 示例代码 2

```
#!/usr/bin/python
import pymongo
mongodbUri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin'
client = pymongo.MongoClient(mongodbUri)
db = client.someonedb
```

```
inserted_id = db.somecoll.insert_one({"somekey":"yiqihapi"}).inserted_id
print inserted_id

for doc in db.somecoll.find(dict(_id=inserted_id)):
    print doc

for doc in db.somecoll.find({"somekey":"yiqihapi"}):
    print doc
```

输出信息，如下所示。

```
5734431e101e2f6d699b37ef
{'u'somekey': u'yiqihapi', u'_id': ObjectId('5734431e101e2f6d699b37ef')}
{'u'somekey': u'yiqihapi', u'_id': ObjectId('5734431e101e2f6d699b37ef')}
```



# Python 读写示例

最近更新时间：2024-01-12 11:04:04

本文以 Python 代码示例来演示 MongoDB 分片集群的数据基本读写操作。该示例代码基于 python 2.7语法编写，版本不同，代码可能存在差异。具体信息，请参见 [MongoDB Python Drivers](#)。

```
#!/usr/bin/python
import pymongo
import random

mongodbUri = 'mongodb://mongouser:1234567a@10.66.153.111:27017/admin'

client = pymongo.MongoClient(mongodbUri)
db = client.test

if 'num' in db.collection_names():
    db.drop_collection('num')

#create database and shardkey, shardkey is name
db_admin=client.admin
db_admin.command('enableSharding', 'test')
db_admin.command('shardCollection', 'test.num', key = {'name':1})

#insert data
print 'insert docs'
db.num.insert_one({'id':1, 'name':'R9', 'des':'pretty'})
db.num.insert_one({'id':2, 'name':'BOY', 'des':'handsome'})
db.num.insert_one({'id':3, 'name':'cat', 'des':'nice'})
db.num.insert_one({'id':4, 'name':'dog', 'des':'clever'})
print 'list all docs'
for i in db.num.find(): print i

#insert update doc
print 'update R9 and delete BOY'
db.num.update_one({"name":"R9"}, {"$set":{"des":"good"}})
db.num.delete_one({"name":"BOY"})
db.num.update_one({"id":3}, {"$set":{"des":"kind"}})

print 'print R9'
for i in db.num.find({"name":"R9"}): print i
print 'list all docs'
for i in db.num.find(): print i
```

运行结果：

```
[root@VM_63_228_centos distribute_test]#  
[root@VM_63_228_centos distribute_test]# python demo.py  
insert docs  
list all docs  
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'pretty', u'id': 1, u'name': u'R9'}  
{u'_id': ObjectId('589c62e99d89702a48ebb10e'), u'des': u'nice', u'id': 3, u'name': u'cat'}  
{u'_id': ObjectId('589c62e99d89702a48ebb10f'), u'des': u'clever', u'id': 4, u'name': u'dog'}  
{u'_id': ObjectId('589c62e99d89702a48ebb10d'), u'des': u'handsome', u'id': 2, u'name': u'BOY'}  
update R9 and delete BOY  
print R9  
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'good', u'id': 1, u'name': u'R9'}  
list all docs  
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'good', u'id': 1, u'name': u'R9'}  
{u'_id': ObjectId('589c62e99d89702a48ebb10e'), u'des': u'kind', u'id': 3, u'name': u'cat'}  
{u'_id': ObjectId('589c62e99d89702a48ebb10f'), u'des': u'clever', u'id': 4, u'name': u'dog'}  
[root@VM_63_228_centos distribute_test]#
```

# Go 连接示例

最近更新时间：2024-01-12 11:04:18

## 相关说明

云数据库 MongoDB 默认提供 `rwuser` 和 `mongouser` 两个用户名，分别支持 MONGODB-CR 和 SCRAM-SHA-1 两种认证方式，对于这两种认证方式，连接 URI 需要做不同的处理，具体请参见 [连接实例](#)。

[mGo 驱动下载](#)、[MongoDB Go 驱动下载](#)

## mGo 驱动示例代码

```
func GetMgoURL(ip, user, password string, port int) string {
    urlString := ""
    if user == "" && password == "" {
        urlString = fmt.Sprintf("mongodb://%s:%d/admin", ip, port)
    } else {
        urlString = fmt.Sprintf("mongodb://%s:%s@%s:%d/admin",
            url.QueryEscape(user), url.QueryEscape(password), ip, port)
    }

    return urlString
}

url := service.GetMgoURL(reqPara.Ip, reqPara.User, reqPara.Password,
    reqPara.Port)
session, err := mgo.Dial(url)
```

## MongoDB Go 驱动示例代码

示例代码请参见 [官方文档](#)。

# PHP 重连示例

最近更新时间：2024-01-12 11:04:32

## 说明

云数据库 MongoDB 提供的不是简单的 mongod 访问，给到用户访问的是一个负载均衡 IP，此 IP 后面是连接到一系列类似 mongos 一样存在的路由接入层。

客户端驱动会透过负载均衡 IP 与接入机建立一个长连接，当此连接处于长期间活跃状态时，我们不会对其做任何干预，但是当长连接闲置时间超过1天时（此时间会随着版本优化而调整），路由接入层会剔除该连接。

一般来说，客户端驱动会实现一个自动重连的过程，但是也有部分语言的驱动并没有实现。对于没有实现自动重连的语言驱动，当用户使用一个已经被剔除的连接来尝试与云数据库 MongoDB 通信时可能会得到“Remote server has closed the connection”之类的错误信息，所以需手动进行重连，本文给出一个 PHP 重连的 demo。

## 基于 php mongo 驱动的重连实现

```
<?php

function getConnection() {
    $connection = false;
    $uri = 'mongodb://rwuser:1234567a@10.66.148.142:27017/admin?
authMechanism=MONGODB-CR';
    $maxRetries = 5;
    for( $counts = 1; $counts <= $maxRetries; $counts++ ) {
        try {
            $connection = new MongoClient($uri);
        } catch( Exception $e ) {
            // 或者 根据需要使用下面的catch代码行，注意那一个“\\”，某些框架使用命名空间时需要用
            // } catch( \\Exception $e ) {
                continue;
            }
            break;
        }
    }
    return $connection;
}

$connection = getConnection();

if($connection) {
```

```
$db = $connection->testdb;
$collection = $db->testcollection;

$one = $collection->findOne();

var_dump($one);
}
```