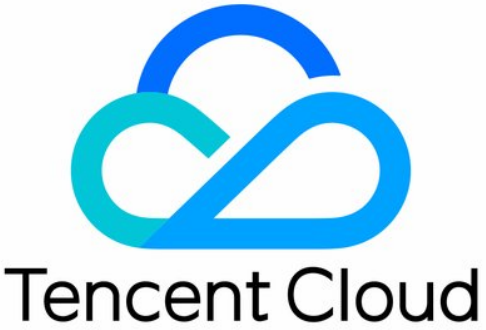
TencentDB for MongoDB SDK Reference



Copyright Notice

Trademark Notice

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

SDK Reference
Shell Connection Example

Last updated: 2025-02-08 11:17:48

You can use the MongoDB shell client (please see the installation documentation) on a CVM instance to connect to TencentDB for MongoDB for data management. Be sure to use the latest version of MongoDB client suite.

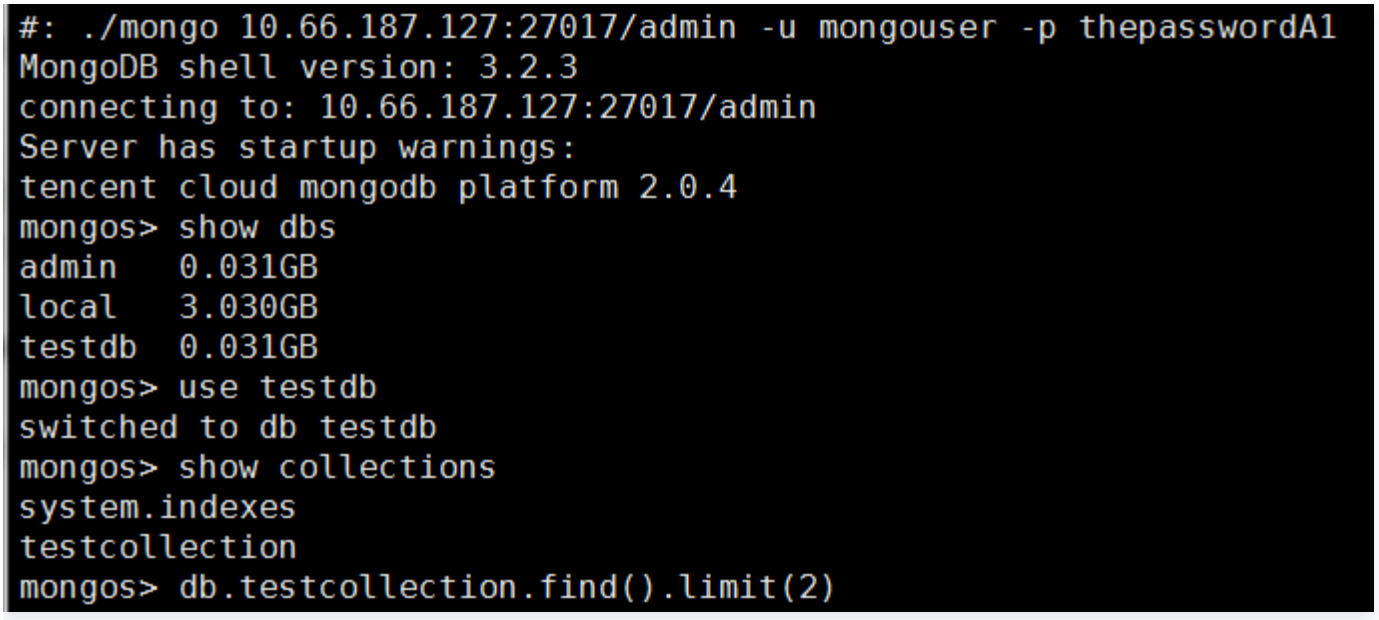
Quick Start

A typical connection command is as follows:

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

> **Note:**
> When using connection string access for MongoDB, if the password contains special characters, they need to be escaped to URL encoding for normal recognition. For example, @ needs to be escaped to %40.

See the figure:



Connection in different authentication methods

In Connect Instances, it is noted that TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support MONGODB-CR and SCRAM-SHA-1

authentication methods respectively. For these two authentication methods, the shell parameters are different. See below for details.

SCRAM-SHA-1 authentication (mongouser)

The default user mongouser and all new users created in the console use SCRAM-SHA-1 authentication. Their shell connection parameters are exactly the same as in the Quick Start section, with no need to add additional parameters. Example as follows:

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

If you want to directly enter a specific db, such as singer, after connecting to the MongoDB service, follow the example operation:

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

As shown in the figure:

```
#: ./mongo 10.66.187.127:27017/singer -u mongouser -p thepasswordA1 --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>
```

MONGODB-CR authentication (rwuser)

Please note that only the default user rwuser uses MONGODB-CR authentication. Its shell connection parameters need to specify the authentication method as MONGODB-CR, example as follows:

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

As shown:

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

If you want to directly enter a specific db, such as singer, after connecting to the MongoDB service, please follow the example:

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

As shown in the figure:

```
#: ./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>
```

Using shell to import and export data

Both authentication methods mentioned above can be used to perform data import and export in the shell. Please refer to Export and Import.

PHP Connection Sample

Last updated: 2025-02-08 11:18:06

Related Description

TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support two authentication methods, MONGODB-CR and SCRAM-SHA-1. For these two authentication methods, the connection URI requires different handling. For details, see Connect Instances.
In PHP, there is a driver that can be used for connection operations with MongoDB database: mongodb (PHP Official Documentation) - MongoDB officially recommends the mongodb driver, but it requires PHP version 5.4 and above.
Below, the above-mentioned driver is used to demonstrate connection to TencentDB for MongoDB and perform read and write.

Using the mongodb Driver

Refer to mongodb installation methods in official installation steps.
The mongodb driver can use MONGODB-CR and SCRAM-SHA-1 authentication methods. For details, see Connect Instances.
Sample code:

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

// Prepare to write data
$document1 = [
    'username' => 'lily',
    'age'      => 34,
    'email'    => 'lily@qq.com'
];

// driver pre-processing data, here you can see that MongoDB's _id is
generated by the driver
$bulk = new MongoDB\Driver\BulkWrite;
$_id1 = $bulk->insert($document1);

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

```
// or use the following code to ensure data writing to most nodes as
needed
// $writeConcern = new
MongoDB\Driver\WriteConcern(MongoDB\Driver\WriteConcern::MAJORITY,
1000);
// $result = $manager->executeBulkWrite('testdb.testcollection', $bulk,
$writeConcern);

// Query
$filter = ['_id' => $_id1];
$query = new MongoDB\Driver\Query($filter);
$rows = $manager->executeQuery('tsdb.table1', $query); // can also
prefer reading from secondary
foreach($rows as $r){
   print_r($r);
}
```

Output:

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

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

Recommending Using the PHPLIB Library (Based on the Mongodb Driver Encapsulation)

It is recommended to use the mongodb driver with PHPLIB, view related documents.
The installation methods for PHPLIB refer to the official installation steps. Please note that PHPLIB depends on the mongodb driver.
Sample code:

```
<?php
```

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

// Initialize
$mongoClient = new
MongoDB\Client('mongodb://mongouser:thepasswordA1@10.66.187.127:27017/ad
min');

// Use the users collection under the demo database
$collection = $mongoClient->demo->users;

// Insert a piece of data
$insertOneResult = $collection->insertOne(['name' => 'gomez']);

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

// Query data.
$document = $collection->findOne(['name' => 'gomez']);

var_dump($document);
```

Output:

```
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 Connection Sample

Last updated: 2025-02-08 11:18:24

Related Description

TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support MONGODB-CR and SCRAM-SHA-1 authentication methods respectively. For these two authentication methods, the connection URI requires different handling. Refer to Connect Instances and Documentation of MongoDB Node.js Driver.

Quick Start

Native Node.js sample code

Install the driver package through shell:

```
npm install mongodb --save
(If the installation failed, you can try another source, such as npm
config set registry http://registry.cnpmjs.org)
npm init
```

Program code:

```
'use strict';

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

// Concatenate the 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'); // Select a database
    var col = db.collection('demoCol'); // Select a collection (table)
   // Insert data.
    col.insertOne(
        {
            a: 1,
            something: "yy"
        },
```

```
        // Optional parameters
        //{
        //    w: 'majority' // Enable the "Majority" mode to ensure that
data are written to the Secondary nodes
        //},
        function(err, r) {
            console.info("err:", err);
            assert.equal(null, err);
            // Assertion is written successfully
            assert.equal(1, r.insertedCount);
            // Query data.
            col.find().toArray(function(err, docs) {
                assert.equal(null, err);
                console.info("docs:", docs);
                db.close();
            });
        }
    );
});
```

Output:

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

Sample Code for Connecting to Node.js mongoose

Mongoose is an object modeling tool for convenient operations on MongoDB in the Node.js asynchronous environment.

Preparations Before Running

Execute the following command to install **mongoose** on the client.

`npm install mongoose --save`

Sample code

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

// Fill in your MongoDB username, password, instance IP address, and
port number in the following parameters
var dbUri = "mongodb://" + user + ":" + password + "@" + host + ":" +
port + "/" + dbName;
```

```
// Establish connection
var opts = {
    auth: {
        authMechanism: 'MONGODB-CR', // This parameter is not needed if
using SCRAM-SHA-1 authentication
        authSource: 'admin'
    }
};
var connection = mongoose.createConnection(dbUri, opts);

// Connection established
mongoose.connection.on('connected', function () {
    console.log('Mongoose connection open to ' + dbUri );
});

// Operate user table (collection)   Define a schema   The objects in
the schema need to correspond one-to-one with the fields in the database
table
const UserSchema = new mongoose.Schema({
    name: String,
    age: Number
});

// Define database model   Database operations
const User = mongoose.model('User', UserSchema, 'user');

// Save data
var addUser = new User({
    name:'James',
    age:36
});

function testSave() {
    addUser.save(function (error, user) {
        console.log("save()", error, user)
    });
}

testSave()
```

Java Connection Sample

Last updated: 2025-02-08 11:18:44

Related Description

TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support two authentication methods, MONGODB-CR and SCRAM-SHA-1. For these two authentication methods, the connection URI requires different handling. For details, see Connect Instances.
Documentation of the MongoDB Java Driver
Java JAR package download, please choose version 3.2 or later.

Quick Start

Native Java sample code

```
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"; //
Multiple IP instances URI can refer to:
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 {
            // Use the database someonedb
            MongoDatabase database =
mongoClient.getDatabase("someonedb");
            // Get the handle of the collection/table someonetable
            MongoCollection<Document> collection =
database.getCollection("someonetable");

            // Prepare to write data
```

```
            Document doc = new Document();
            doc.append("key", "value");
            doc.append("username", "jack");
            doc.append("age", 31);

            // Write data
            collection.insertOne(doc);
            System.out.println("insert document: " + doc);

            // Read data
            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 {
            // Close the connection
            mongoClient.close();
        }
    }
}
```

Output:

```
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.
```

Configuration sample for Spring Data MongoDB

This sample demonstrates how to configure the authentication database admin, which depends on the versions of Spring and Spring Data MongoDB you use.

```
<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="your target database" />
    <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="username" />
    <constructor-arg name="password" value="password" />
</bean>
```

Python Connection Sample

Last updated: 2025-02-08 11:19:05

Related Description

- TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support two authentication methods, MONGODB-CR and SCRAM-SHA-1. For these two authentication methods, the connection URI requires different handling. For details, see Connect Instances.
- Download Python Driver. For specific information, see MongoDB Python Drivers.

Quick Start

Python sample code 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 sample code 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
```

Output information, as shown below.

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

Python Read/Write Sample

Last updated: 2025-02-08 11:40:01

This document uses Python sample code to demonstrate the basic data read/write operations in a TencentDB for MongoDB sharded cluster. The sample code is written based on Python 2.7 syntax, and the code may vary in different versions. For specific information, see 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
```

Execution result:

```
[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'}
```

Go Connection Sample

Last updated: 2025-02-08 11:19:50

Related Description

TencentDB for MongoDB default provides two usernames, rwuser and mongouser, which support MONGODB-CR and SCRAM-SHA-1 authentication methods respectively. For these two authentication methods, the connection URI requires different handling. For details, please refer to Connect Instances.
mGo Driver Download, MongoDB Go Driver Download

Sample Code for mGo Driver

```
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)
```

Sample Code for MongoDB Go

For sample code, please refer to the official documentation.

PHP Reconnection Sample

Last updated: 2025-02-08 11:20:31

Description

Instead of simply allowing you to access mongod, the TencentDB for MongoDB database service provides a load balancer IP for access. You can use this IP to connect to a range of route access layers similar to mongos.
The client driver establishes a persistent connection with an access server through the load balancer IP. If the connection is active for a long period of time, no intervention will be imposed on this status. However, if the persistent connection is inactive for more than one day (this period will be adjusted with version optimization), the route access layer will terminate the connection.
Generally, the client driver will implement an automatic reconnection process. However, this process cannot be implemented by certain language drivers. For such language drivers, if you attempt to communicate with the TencentDB for MongoDB service through a terminated connection, an error message such as "Remote server has closed the connection" will be returned, and manual reconnection will be required. The document provides a demo for PHP reconnection.

Reconnection Based on PHP Mongo Driver

```
<?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 ) {
     // Or use the following catch code line as needed, note the "\",
which is required by some frameworks when using namespaces.
     // } catch( \Exception $e ) {
            continue;
        }
        break;
    }
    return $connection;
```

```
}

$connection = getConnection();

if($connection) {
        $db = $connection->testdb;
        $collection = $db->testcollection;

        $one = $collection->findOne();

        var_dump($one);
}
```