



Driver Authentication

Abstract

MongoDB supports various authentication strategies across various versions. When authentication is turned on in the database, a driver must authenticate before it is allowed to communicate with the server. This spec defines when and how a driver performs authentication with a MongoDB server.

This report is following a specification from the [Mongodb](#) website and is shown below.

Mongo credential

Drivers should contain a type called MongoCredential. In this driver it is called `MongoDB::Authenticate::Credential`.

```
use lib 't';
5. use Test-support;
   use MongoDB;
   use MongoDB::Client;
   use MongoDB::Database;
   use MongoDB::HL::Users;
10. use MongoDB::Authenticate::Credential;
    use BSON::Document;
    set-logfile($*OUT);
    set-exception-process-level(MongoDB::Severity::Trace);
    my MongoDB::Test-support $ts .= new;
15. my MongoDB::Client $client;
    my MongoDB::Authenticate::Credential $cred .= new(
        :username<user>, :password<pencil>,
        :auth-mechanism<SCRAM-SHA-1>
    );
20. ok $cred.defined, 'T0';
    is $cred.password, "pencil", 'T1';
```

- ✓ **T0:** Credential is defined. Not all named arguments are used. There are also `:source` and `:mechanism-properties`. Btw. the field `mechanism` is named `auth-mechanism` because of the name `authMechanism` used in the url as an option.
- ✓ **T1:** Data can also be retrieved from it again.

Some combinations of credential parameters should fail

```

my $x = {
  $cred .= new(
    :username<user>, :password<pencil>, :auth-mechanism<MONGODB-X509>
25. );
};
dies-ok $x, 'T2';
my $y = {
  $cred .= new( :password<pencil>, :auth-mechanism<MONGODB-CR>);
30. };
dies-ok $y, 'T3';

```

✓ **T2:** Mechanism **MONGODB-X509** does not expect a password

✓ **T3:** Mechanism **MONGODB-CR** must have a username

Credential parameters are taken from the information in the uri. The authentication mechanism can also be guessed from the max wire version from the info returned by the `isMaster` command. This, however can only be checked after a socket has been opened caused by a user action. B.t.w Cleanup the client object when done because its concurrent processes will keep connected to the server even after undefining. The driver will not show much if no user or password is given. So, in order to test that a user must be created. All of this can only be done when the server is not yet in authentication mode or, it is, and one can login using an administrative account. We will create an account `user` with password `pencil` . It gets read write rights to the database `tdb`.

```

my Int $port1 = $ts.server-control.get-port-number('s1');
my MongoDB::Database $database = $client.database('tdb');
my MongoDB::HL::Users $users .= new($database);
35. $client .= new(uri("mongodb://localhost"));
my BSON::Document $doc = $users.create-user(
  'user', 'pencil',
  :roles([(role => 'readWrite', db => 'tdb'),])
);
40. ok $doc<ok>, 'T4';

```

☆☆ **T4:** User `user` created

Now we can try to login using the credentials

```

$client .= new(uri("mongodb://user:pencil@localhost:$port1/tdb"));
ok $client.defined, 'T5';
is $client.credential.auth-mechanism, "", 'T6';
$doc = $database.run-command: (
45.   insert => 'tcol',
      documents => [
        BSON::Document.new: (
          name => 'Larry',
          surname => 'Walli',
50.       ),
      ]
);
say "Doc:", $doc.perl;
is $doc<ok>, 1, 'T7';
55. is $doc<n>, 1, 'T8';
is $client.credential.auth-mechanism, "SCRAM-SHA-1", 'T9';
$client.cleanup;

```

- ☆☆ **T5:** Returned client object is defined, even when the uri is not pointing to an existing mongod server.
- ☆☆ **T6:** Mechanism is not yet defined.
- ☆☆ **T7:** insert request ok
- ☆☆ **T8:** inserted 1 document
- ☆☆ **T9:** Mechanism is a SCRAM-SHA-1 version

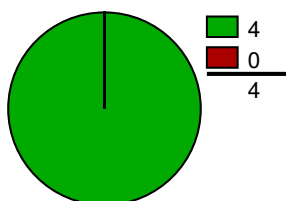
```

$client .= new(uri("mongodb://localhost:$port1"));
ok $client.defined, 'T10';
60. $client.cleanup;

```

- ☆☆ **T10:** Returned client object is defined, even when the uri is not pointing to an existing mongod server.

Normal tests



Bug issues



Todo tests



Summary

